Project no.:	**027657**
Project full title:	**Perception, Action & Cognition through learning of Object-Action Complexes**
Project Acronym:	**PACOPLUS**
Deliverable no.:	**D4.2.1**
Title of the deliverable:	**The Integration of Objects and Action Plans**

Contractual Date of Delivery to the CEC: **31 July 2007**

Actual Date of Delivery to the CEC: **31 July 2007**

Organisation name of lead contractor for this deliverable: UL

Author(s): Joyca Lacroix, Bernhard Hommel, Justus Piater, Kai Hübner, Danica Kragic, Tamim Asfour, Kai Welke, Norbert Krüger, Dirk Kraft.

Participants(s): UL, UniKarl, KTH, BCCN, JSI, SDU, ULg

Work package contributing to the deliverable: WP 4 (and parts of WP 1, WP 2, WP 6, WP 7)

Nature: R

Version: 1.0

Total number of pages: 6

Start date of project: 1 Feb 2006 **Duration: 48 months**

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:

The core focus of WP4 is the formalization of Object Action Complexes (OACs). As part of WP4.2 this deliverable reports on the development of representational systems for Object-Action Complexes that integrate sensory attributes and action attributes acquired from various types of active behaviors, e.g., oculomotor, grasping, moving, and approaching behaviors. This deliverable comprises five papers that present our work on representational systems for the integration of perception and action. The proposed methods address the construction of object models that support visual detection, recognition, and classification of objects in realistic/natural scenes, as well as their robotic manipulation.

Keyword list: Representational system, integration of objects and action

Table of Contents

1. Executive Summary

The core focus of WP4 is the formalization of Object Action Complexes (OACs). This deliverable is part of WP 4.2 and particularly addresses the development of a representational system for Object-Action Complexes (OACs). The main idea of an OAC is that it integrates various types of sensory and action attributes acquired from interactive exploration of the environment. At this early stage of the duration period of this work package (formally started in month 13), various approaches and possible cooperations are discussed that aim at combining efforts across work package groups in order to obtain a unified representational system that represents the sensory and action features from a wide range of active behaviors, e.g., oculomotor, grasping, moving, and approaching behaviors. The representational system that aims at supporting the efficient interaction of the robot with the environment (1) builds on work on multisensorial attributes performed in the context of WP 4.1 and (2) communicates with the LDEC system developed in the context of WP 4.3 for the recognition and planning of goals. So far, several important steps have been made, which contribute to the development of an integrated representational system that combines sensory and action information acquired from perceiving and operating on a realistic environment. The operating behaviors employed in our studies involve mainly oculomotor and grasping behaviors.

This deliverable comprises five papers that present first results of our research on representational systems designed to link perception and action. The proposed methods address the construction of object models for visual detection, recognition, and classification of objects in realistic/natural scenes, as well as their robotic manipulation. Below, we briefly sketch the contribution of each paper to this work package.

[A] *(Presented at Robotics, Science, and Systems 2007 in Atlanta, GA, USA).* This paper presents an object representation system that builds hierarchical object models on the basis of primitives localized in Euclidean space by position and orientation. Models are constructed by detecting stable spatial relations of these primitives when viewing a scene, and representing these relations in a probabilistic graphical model. The learned models can be directly used for object detection, recognition and pose estimation; the paper provides a systematic evaluation of the accuracy of estimated poses. This framework permits the seamless inclusion of non-visual features such as kinematic parameters, and will later be used to learn associations between visual and kinematic features for robotic grasping.

[B] *(Submitted and partly presented at Humanoids-06 in Genoa, Italy).* This paper presents an initial step toward a visual cognitive system that is inspired by active saccadic human vision to build visual representations. The system combines a perceptual mechanism with an episodic memory system for the active exploration of an object by means of saccades. The perceptual mechanism builds visual representations that are acquired by saccadically fixating different parts of the visual scene. The episodic memory system (i.e., the representational system) stores the saccadic actions and the resulting sensory input and uses this episodic knowledge to direct saccades to spatial locations that are relevant for the classification/recognition of a newly encountered object. As such it adheres to neurophysiological and psychological insights about the use of bottom-up and top-down driven processes in active human vision.

[C] *(Draft paper to be submitted during the next months).* This paper presents an intuitive approach towards grasping unknown objects in whole or by parts. One important aim of the project is founded in providing a robot actuator system with a set of primitive actions, e.g. pick-up an arbitrary object from a table. For performing such basic actions, it is most valuable to model the object from 3D sensory input. However, we have to state the question up to which detail this is necessary. Complex shape approximations are difficult to process, while simple ones will give worse approximation. We prefer general fast on-line techniques instead of pre-learned off-line examples, thus the algorithm's efficiency is the more important. Unknown objects are hardly parametrizable but need real-time application for robot grasping. We adopted those motivations to develop a novel approach using boxes as a mid-level representation. In our approach, we combine different incentives on simplicity of boxes, efficiency of hierarchies and fit-and-split algorithms for shape approximation in terms of grasping.

[D] *(Accepted for the 2007 IEEE International Conference on Intelligent Robots and Systems).* In this paper an approach to represent objects with respect to the similarity of their appearance is presented. The proposed appearance based approach motivates a multi-modal representation scheme of objects. Once multiple modalities are combined into one unique percept, the storage requirements for each modality can be considerably reduced by exploiting similarities of objects and object views. This reduction allows acquiring and storing larger amounts of object representations while preserving the ability to recognize objects.

[E] *(Submitted and partly presented at two workshops at VISAPP 2007 and ICRA 2007).* In this paper, a procedural definition of the detection of ‘objectness’ as well as an algorithm to extract the object shape is given by the interaction of two OACs. The first OAC is a ‘grasping reflex’ by which physical control over unknown objects can be achieved. The second OAC becomes triggered in case that the first OAC is successful. It extracts object shape by making use of object motion induced by the robot. By these two modules, the existence of objects in the scene becomes detected (‘Birth of the object’) which is then information that is relevant for the planning side.

Together the papers present a number of important contributions to the development of an OAC-based representational system for the support of efficient active behaviors in a realistic environment:

- A probabilistic framework for learning cross-modal object models that associate object appearance and pose with grasp parameters.
- A visual representational system that is informed by active human vision; it gathers visual input by means of a sequence of saccades and uses stored representational knowledge to guide the selection of relevant visual input in a top-down manner.
- A system that approaches intuitive grasp hypotheses generation for the grasping of unknown objects. Shape approximating sets of oriented bounding boxes are used as a key for efficient grasp hypotheses generation and refinement as also for including task dependency.
- Generation of compact appearance-based object representations which preserve the similarity among objects and object view.
- A procedural definition of objects by means of two interacting OACs that produces information that can be passed to the high level planning module.

This work package also relies heavily on and informs a number of other work packages, including WP 1, WP 2, WP 6 and WP 7. The development of a representational scheme is influenced by neurophysiological and cognitive ideas also reflected in the cognitive architecture examined in WP 1. Moreover, the idea of a representational scheme acquired from and used for the support of active behaviors, directly relates to the imitation learning, the machine learning and the perception-action themes studied in the context of WP 2, WP 6 and WP 7, respectively.

2. Attached Papers

A. Hierarchical Integration of Local 3D Features for Probabilistic Pose Recovery

Renaud Detry and Justus Piater

Presented and published in the Proceedings of the Robotics, Science, and Systems Workshop: Robot Manipulation: Sensing and Adapting to the Real World, 2007.

Abstract: This paper presents a 3D object representation framework. We develop a hierarchical model based on probabilistic correspondences and probabilistic relations between 3D visual features. Features at the bottom of the hierarchy are bound to local observations. Pairs of features that present strong geometric correlation are iteratively grouped into higher-level meta-features that encode probabilistic relative spatial relationships between their children. The model is instantiated by propagating evidence up and down the hierarchy using a Belief Propagation algorithm, which infers the pose of high-level features from local evidence and reinforces local evidence from globally consistent knowledge. We demonstrate how to use our framework to estimate the pose of a known object in an unknown scene, and provide a quantitative performance evaluation on synthetic data.

B. Toward a Visual Cognitive System using Active Top-down Saccadic Control

Joyca Lacroix, Eric Postma, Jaap van den Herik, and Jaap Murre

Submitted (Part of this work was presented and published in the Proceedings of the Humanoids-06 Workshop: Toward Cognitive Humanoid Robots, 2006).

Abstract: The saccadic selection of relevant visual input for preferential processing allows for the efficient use of computational resources. Based on saccadic active human vision, we aim to develop a plausible saccade-based visual cognitive system for a humanoid robot. This paper presents two initial steps toward our objective by extending the saccade-based memory model called NIM to a plausible model of natural visual classification. As a first step, we adapt NIM to a straightforward saccade-based model for the classification of natural visual input called NIM-CLASS and evaluate the model in a face-classification experiment. As a second step we aim to approach the interactive nature of human vision by extending NIM-CLASS to NIM-CLASS^{TD} by adding active top-down saccadic control. We then assess to what extent top-down control enhances the performance on the classification task. The results show that the incorporation of top-down saccadic control benefits classification performance compared to the purely bottom-up control, reducing the amount of visual input required for correct classification. Our results lead us to the conclusion that NIM-CLASS^{TD} may provide a fruitful basis for an active visual cognitive system for a humanoid robot that allows for the efficient use of the robot's processing resources.

C. Minimum Volume Bounding Box Decomposition for Robot Grasping

Kai Huebner, Johan Sommerfeld, Steffen Ruthotto and Danica Kragic.

Draft paper to be submitted during the next months.

Abstract: Thinking about intelligent robots involves consideration of how such systems can be enabled to perceive, interpret and act in arbitrary and dynamic environments. While sensor perception and model interpretation focus on the robot's internal representation of the world rather passively, robot grasping capabilities are needed to actively execute tasks, modify scenarios and thereby reach versatile goals. These capabilities should also include the generation of stable grasps to safely handle even objects unknown to the robot. We believe that the key to this ability is not to select a good grasp depending on the identification of an object (e.g. as a cup), but on its shape (e.g. as a composition of shape primitives). In this paper, we envelop given 3D data points into primitive box shapes by a fit-and-split algorithm that is based on an efficient Minimum Volume Bounding Box implementation. Though box shapes are not able to approximate arbitrary data in a precise manner, they give efficient

clues for planning grasps on arbitrary objects. We present the algorithm and experiments using the 3D grasping simulator GraspIt!

D. Exploiting Similarities for Robot Perception

Kai Welke, Erhan Oztop, Gordon Cheng, and Rüdiger Dillmann.
Accepted for the 2007 IEEE International Conference on Intelligent Robots and System.

Abstract: A cognitive robot system has to acquire and efficiently store vast knowledge about the world it operates in. To cope with every day tasks, the robot needs to learn, classify and recognize a manifold of different objects. Our work focuses on an object representation scheme that allows storing perceived objects in a compact way. This will enable the system to store extensive information about the world and will ease complex recognition tasks. The human visual system deploys several mechanisms to reduce the amount of information. Our goal is to develop an artificial system that mimics these mechanisms to create representations that can be used in cognitive tasks. In particular, in this paper we will present our approach that exploits similarities among different views of objects. The proposed representation scheme allows for reduction of storage required for the representation of objects and preserves the information about the similarity among objects. This is achieved by selecting 'important views' of objects, depending on their stability. Furthermore, by extending the same approach to multiple objects, we are able to exploit similarities between objects to find a common representation and to further reduce the storage requirements.

E. Birth of the Object: Detection of Objectness and Extraction of Object Shape through OACs

Dirk Kraft, Emre Baseski, Mila Popovic, Norbert Kruger, Nicolas Pugeault, Danica Kragic, Sinan Kalkan, and Florentin Worgotter
Submitted.

Abstract: We describe a process in which the segmentation of objects as well as the extraction of the object shape becomes realized through active exploration of a robot vision system. In the exploration process, two behavioural modules that link robot actions to the visual and haptic perception of objects interact. First, by making use of an object independent grasping mechanism, physical control over potential objects can be gained. Having evaluated the initial grasping mechanism as being successful, a second behaviour extracts the object shape by making use of prediction based on the motion induced by the robot. This also leads to the concept of an 'object' as a set of features that change predictably over different frames. The system is equipped with a certain degree of generic prior knowledge about the world in terms of a sophisticated visual feature extraction process in an early cognitive vision system, knowledge about its own embodiment as well as knowledge about geometric relationships such as rigid body motion. This prior knowledge allows for the extraction of representations that are semantically richer compared to other approaches.

Hierarchical Integration of Local 3D Features for Probabilistic Pose Recovery

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

Justus Piater
Montefiore Institute, University of Liège, Belgium
Email: Justus.Piater@ULg.ac.be

Abstract— **This paper presents a 3D object representation framework. We develop a hierarchical model based on probabilistic correspondences and probabilistic relations between 3D visual features. Features at the bottom of the hierarchy are bound to local observations. Pairs of features that present strong geometric correlation are iteratively grouped into higher-level meta-features that encode probabilistic relative spatial relationships between their children. The model is instantiated by propagating evidence up and down the hierarchy using a Belief Propagation algorithm, which infers the pose of high-level features from local evidence and reinforces local evidence from globally consistent knowledge. We demonstrate how to use our framework to estimate the pose of a known object in an unknown scene, and provide a quantitative performance evaluation on synthetic data.**

I. Introduction

Objects can be characterized by configurations of parts. This insight is reflected in computer vision by the increasing popularity of representations that combine local appearance with spatial relationships [1, 2, 12]. Such methods are richer and more easily constructed than purely geometric models, more expressive than methods purely based on local appearance such as bag-of-features methods [10, 3] and more robust and more easily handled in the presence of clutter and occlusions than methods based on global appearance. Moreover, they not only allow bottom-up inference of object parameters based on features detected in images, but also top-down inference of image-space appearance based on object parameters.

We have recently presented a framework for unsupervised learning of hierarchical representations that combine local appearance and probabilistic spatial relationships [13, 14]. By analyzing a set of training images, our method creates a codebook of features and observes recurring spatial relationships between them. Pairs of features that are often observed in particular mutual configurations are combined into a meta-feature. This procedure is iterated, leading to a hierarchical representation in the form of a graphical model with primitive, local features at the bottom, and increasingly expressive meta-features at higher levels. Depending on the training data, this leads to rich representations useful for tasks such as object detection and recognition from 2D images.

We are currently developing an extension of this method to 3D, *multi-modal* features. We intend to integrate multiple perceptual aspects of an object in one coherent model, by combining visual descriptors with haptic and proprioceptive information. This will be directly applicable to robotic tasks such as grasping and object manipulation. Correlated percepts of different natures will induce cross-modal associations; a grasp strategy may be linked directly to visual features that predict its applicability.

In this paper, we focus on hierarchical models for visual object representation. Here, an *observation* is an oriented patch in 3–space, annotated by various visual appearance characteristics. To infer the presence of an object in a scene, evidence from local features is integrated through bottom-up inference within the hierarchical model. Intuitively, each feature probabilistically votes for all possible object configurations consistent with its pose. During inference, a consensus emerges among the available evidence, leading to one or more consistent scene interpretations. The system never commits to specific feature correspondences, and is robust to substantial clutter and occlusions.

We illustrate our method on the application of object pose estimation. Object models are learned within a given world reference frame, within which the object is placed in a reference pose. Comparing an instance of the model in an unknown scene with an instance in the learned scene allows us to deduce the object pose parameters in the unknown scene.

II. 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. Figure 1 shows a didactic example of a hierarchy for a bike. The bike is the composition of *frame* and *wheel* features. A wheel is composed of pieces of tire and spokes. The generic piece of tire at the

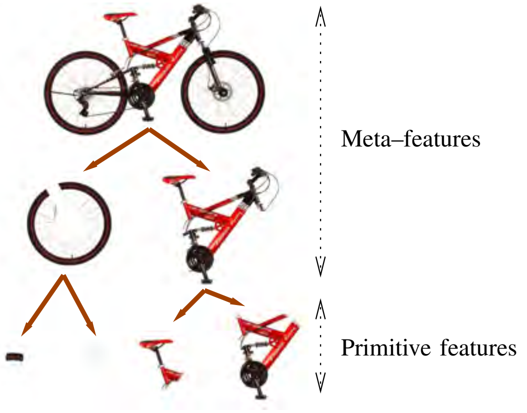


Fig. 1. A didactic example of a hierarchical model of a bike.

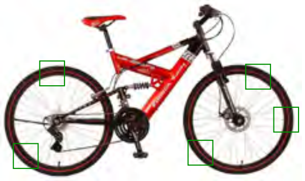


Fig. 2. Instances of the generic piece-of-tire primitive feature in the bike scene.

bottom of the hierarchy is a primitive feature; the pieces of tire squared in green in the scene (Figure 2) are instances of that primitive feature.

At the bottom of the hierarchy, primitive features 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 the feature observations, by associating observations to primitive features.

In summary, information about an object is stored within the model in the three following forms:

i. the topology of the hierarchy,
ii. the relationships between related features,
iii. the codebook vectors annotating bottom-level features.

A. Parametrization

Formally, the hierarchy is implemented using a Pairwise Markov Random Field (see Figure 3). Features are associated to hidden nodes (white in Figure 3), and the structure of the hierarchy is reflected by the edge pattern between them. Each meta-feature is thus linked to its two child features. Observed variables y_i of the random field stand for observations.

When a model is associated to a particular scene (during construction or instantiation), features are associated to corresponding instances in that scene. The correspondence between a feature i and its instances is represented by a probability density over the pose space $SE(3) = \mathbb{R}^3 \times SO(3)$ represented by a random variable x_i.

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 $SE(3)$.

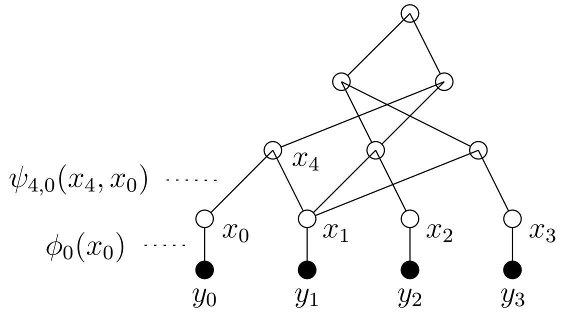


Fig. 3. A Pairwise Markov Random Field representing a feature hierarchy. Features correspond to hidden variables (white). Observed variables (black) correspond to observations, bound to bottom-level primitive features.

Compatibility potentials allow relationship distributions to have multiple modes. In the bike model, let us consider the meta-feature that represents a generic wheel. There are two wheels in the picture; two instances of the wheel feature will be used in a mental reconstruction of the bike. Hence, the compatibility potential between the *wheel* feature and the *bike* feature will be dense around two modes, one corresponding to the transformation between the bike and the front wheel ("the front wheel is on the right side of the bike"), the other between the bike and the rear wheel ("the rear wheel is on the left side of the bike").

Finally, 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 x_i's observations.

The term *primitive feature instance* formally refers to a random draw from a primitive feature distribution. While a primitive feature instance often corresponds to an observation, observations enter into the graphical model merely as prior knowledge. Primitive feature instances result from inference; they depend on observations *and* on all features of the hierarchy. Owing to inference mechanisms presented in the next paragraph, if an observation is discarded (e.g. occluded), a primitive feature instance may nevertheless appear at its place.

B. Model Instantiation

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. Thus, once priors (observation potentials, evidence) have been defined, instantiation can be achieved by any applicable inference algorithms. We currently use a Belief Propagation algorithm described in Section III-A.

For primitive features, evidence is estimated from feature observations. Observations are classified according to the primitive feature codebook; for each primitive feature i, its observation potential $\phi_i(x_i)$ is estimated from observations that are associated to the i^{th} codebook vector. For meta-features, evidence is uniform.

C. Model Construction

The construction procedure starts by clustering feature observations in the appearance space to build a codebook of observations. The number of classes is a parameter of the system. These classes are then used to initialize the first level of the graph:

1) A primitive feature is created for each class;
2) Each primitive feature is tagged with the codebook vector (cluster center) of its corresponding class.

The spatial probabilistic density of each primitive feature is then computed from the spatial distribution of corresponding observations. We use nonparametric representations (see section III-B); the set of observations bound to each primitive feature can thus be directly used as a density representation.

After primitive features have been computed, the graph is built incrementally, in an iterative manner. The construction algorithm works by extracting feature co-occurrence statistics. Features that tend to occur at non-accidental relative positions are repeatedly grouped into a higher-level meta-feature. At each step, the top level of the graph is searched for strongly correlated pairs of features. The k most strongly correlated pairs are selected to form the k meta-features of the next level. The number of meta-features created at each step is a parameter, which we usually keep equal to the initial number of classes. The search for strong feature combinations is the operation responsible for the *topology* of the graph.

The k new meta-features are then provided with a spatial probability distribution, generated from a combination of the children's densities. The meta-feature is placed in the middle of its children, location- and orientation-wise (thus, the meta-feature distribution will be dense between dense regions of the children's distributions). Finally, spatial relations between each meta-feature and its children are extracted, which defines the compatibility potentials. This is achieved by repeatedly taking a pair of samples, one from the parent distribution and one from a child's distribution. The spatial relationships between a large number of these pairs form the relationship distribution between the parent and that child. While the search for strong combinations was responsible for the topology of the graph, the extraction of spatial relations is responsible for the *parametrization* of the graph through the definition of compatibility potentials associated with edges between adjacent features. This parametrization constitutes the principal outcome of the learning algorithm. Relationship extraction is the last operation of a level-construction iteration.

Incremental construction of the graph can, in principle, continue indefinitely, growing an ever-richer representation of the observed scene. The number of levels is a parameter that is chosen to reach a desired level of abstraction; its effect will be discussed in Section V.

III. Implementation

A. Inference

Graphical models are a convenient substrate of sophisticated *inference algorithms*, i.e. algorithms for efficient computation of statistical quantities. An efficient inference algorithm is essential to the hierarchical model, for it provides the mechanism that will let features communicate and propagate information.

Our inference algorithm of choice is currently the Belief Propagation algorithm (BP) [11, 16, 6]. Belief Propagation is based on incremental updates of marginal probability estimates, referred to as *beliefs*. The belief at feature i is denoted

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

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 $m_{ij}(x_j)$, 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

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

where Z is a normalizing constant. To prepare a message for feature j, feature i starts by computing a local "pose belief estimation", 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

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

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 marginal pose densities at top-level features. 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.

B. Nonparametric Representation

We opted for a nonparametric approach to probability density representation. A density is simply represented by a set of particles; the local density of these particles in space is proportional to the actual probabilistic density in that region. Compared to usual parametric approaches that involve a limited number of parametrized kernels, problems like fitting of mixtures or the choice of a number of components can be avoided. Also, no assumption concerning the shape of the density has to be made.

Particles live in the Special Euclidean Space $SE(3)$. The location/translation component is parametrized by a 3–vector. For the orientation/rotation component it was decided to prefer quaternions over rotation matrices, for they provide a well-suited formalism for the manipulation of rotations such as composition or metric definition [9, 7].

For inference, we use a variant of BP, Nonparametric Belief Propagation, which essentially develops an algorithm for BP message update (2) in the particular case of continuous, non-Gaussian potentials [15]. The underlying method is an extension of particle filtering; the representational approach is thus nonparametric and fits our model very well.

IV. Object Pose Estimation

Since features at the top of an object model represent the whole object, they will present relatively concentrated densities that are unimodal if exactly one instance of this object is present in the scene. These densities 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 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. For every top-level feature i of the instantiated graph, compute a *reference aggregate feature pose* π_1^i from its unimodal density.
   Instantiating the model in a reference scene is necessary because even though the top-level features all represent the whole object, they come from different recursive combinations of features of various poses.
2) Instantiate the object model in the unknown scene. For every top feature of that graph, compute an *aggregate feature pose* π_2^i.
3) For all top level features i, the transformations from π_1^i to π_2^i should be very similar; let us denote the mean transformation t. 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.

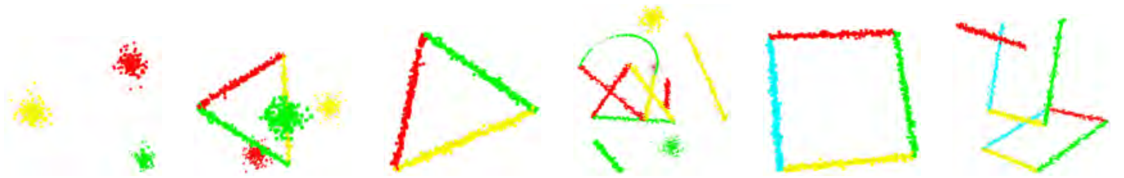
V. Experiments

We ran pose estimation experiments on a series of artificial “objects” presented in Figure 4. In these experiments, we bypass the clustering step and directly generate evidence for primitive features. Since we use nonparametric density representations, we generate observations that directly become evidence for primitive features. Primitive features may have distributions in the shape of blobs, lines, and curves (see Figure 4). For a blob, location components of observations are drawn from a Gaussian distribution around a random 3D point; orientation components are drawn from a Von Mises-Fisher distribution [5, 4] centered at a random 3D orientation. For a line, locations are drawn from a Gaussian distribution around a line segment; orientations are drawn from a Von Mises-Fisher distribution centered at a 3D orientation such that its main direction is along the line and its second direction is in a fixed plane. Figure 5 illustrates orientations.

In the next paragraphs, we go through the procedure of a pose estimation experiment. First, a model is learned from one set of observations of an object of interest (the reference scene). A hierarchy is built up to n levels, we instantiate the model in the reference scene, and compute a reference aggregate feature pose π_1^i for every top feature i of the model.

We are then ready to estimate the pose of our object in a novel, noisy scene. We initialize primitive-feature evidence of the model on a fresh draw of observations of the object of interest in a random pose *plus* observations of a foreign object (see Figures 4(b), 4(d), 4(f)). Evidence is propagated through the hierarchy, and we can eventually estimate the top-feature poses. Since the object of interest is present only once in the noisy scene, top level features should, after instantiation, present unimodal densities; we can safely compute a mean pose π_2^i for each of them.

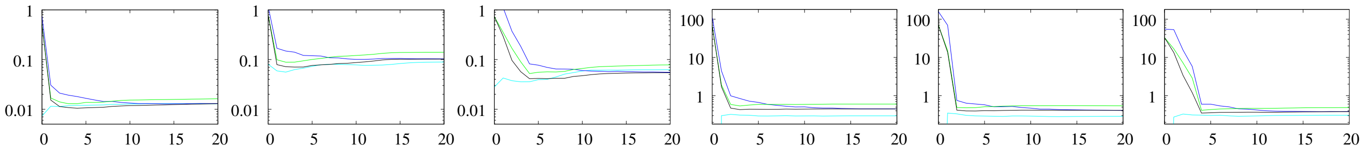
Finally, we compute the transformation t_i between π_1^i and π_2^i for every top feature i. As noted in Section IV, all t_i are very similar. Let us denote the mean transformation t, which corresponds to the *estimated* rigid body motion between the pose of the object in the reference scene, and its pose in the noisy scene. Let us also denote δt the standard deviation of individual t_i's around t.



(a) **blobs** (object) (b) **blobs** (noisy scene) (c) **triangle** (object) (d) **triangle** (noisy scene) (e) **square** (object) (f) **square** (noisy scene)

Fig. 4. Synthetic object observations in Figures (a), (c), (e); noisy scene for each object in Figures (b), (d), (f). Each figure shows primitive-feature densities; color indicates the different primitive feature classes. For instance, Figure (a) shows a simple object consisting of three blobs. The bottom level of the hierarchy corresponding to this object will be composed of three primitive features. For each blob, all observations are associated to one and the same primitive feature.

Fig. 5. Artificially generated observations and their poses.



(a) Translation error (blobs) (b) Translation error (triangle) (c) Translation error (square) (d) Rotation error (blobs) (e) Rotation error (triangle) (f) Rotation error (square)

Fig. 6. Error of the translation (relative to object size) and rotation estimates (in degrees) as a function of the number of levels. The cyan line indicates the error when the pose is estimated on a background-noise–free scene, i.e. for an experiment similar to that described in the text, except that we do not add observations from a foreign object before pose estimation. This error is already low at level 0, since the mean of each primitive feature observations for model instantiation is very similar to that used for model learning. The black line indicates the mean error for noisy scenes – i.e. scenes including foreign objects. The green and blue lines indicate the variance across runs and across top-level nodes. See the text for details.

To evaluate the quality of our estimation, we compare t to the *ground truth* rigid body motion T of the object of interest between the reference scene and the noisy scene. Comparison relies on the distances between translations and distances between rotations. The distance between two rotations θ and θ' is defined as the angle (in degrees) of the 3D rotation that moves from θ to θ'. It can be computed using the quaternion representations of θ and θ' as [9]:

$$\mathbf{d}(q, q') = 2 \arccos(|q \cdot q'|).$$

For each object, this experiment is repeated with different hierarchy heights, from 0 to 20, and for different random seeds. Results are presented in Figure 6. Let us denote $(\lambda_t^s, \theta_t^s)$ and $(\lambda_T^s, \theta_T^s)$ the translational and rotational parts of transformations t and T for a random seed s. Figures 6(a), 6(b) and 6(c) show the mean error of translation estimates as a function of the number of levels. They present on a logarithmic scale the mean distance between λ_t^s and λ_T^s for all s divided by the global size of the object. The global size of the object is defined as the standard deviation of its observations from its center of gravity. Figures 6(d), 6(e) and 6(f) show, on a logarithmic scale, the mean error in degrees of rotation estimates as a function of the number of levels.

The mean error is always large for shallow hierarchies, but decreases rapidly for taller hierarchies until it eventually reaches a stable value. For objects of increasing complexity, this happens at increasingly higher levels. In particular, the noisy scene for the square contains the square itself, plus a second shape that corresponds to a square with one displaced edge. It is only after level 4 that the wrong shape is discarded, and a correct pose of the square is successfully estimated. The triangle has to be detected in a very noisy scene. This leads to a larger translational error that does not get smaller than 0.1 – about 5% of the edge length of the triangle.

In Figure 6, green lines give an idea of the variance between runs under different random seeds. They show the mean error plus three standard deviations. This variance is relatively large since the random variations affect both the synthetic scenes and the models constructed. Lines in blue show the mean error plus three times the mean (over individual runs) of inter-feature standard deviations δt; they give an idea of the variance between top-level features of the same graph during a given run. This variance is large for shallow hierarchies, but converges to 0 for higher levels, which means that top-level features of a model tend to agree more and more as we use taller hierarchies.

The accuracy of pose estimation is further illustrated in Figure 7 that shows the noisy triangle scene (green) and the estimated triangle pose (red).

In the above experiments, feature observations are generated synthetically. Thereby, we avoid the problem of extracting 3D features from sets of images. By manually associating observations to primitive features, we have control over the clustering step. Since the features are synthesized in 3D, there are no viewpoint issues. Despite their simplicity, these experiments demonstrate the feasibility of our sophisticated method.

One way to obtain 3D feature observations from real objects is the early-cognitive-vision system MoInS [8], which extracts 3D primitives from stereo views of a scene (see Figure 8).

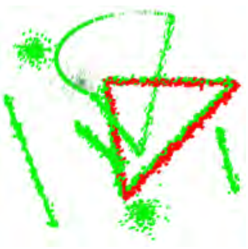
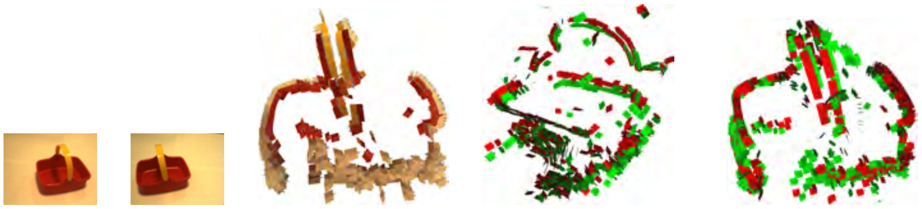


Fig. 7. Accuracy of pose estimation. The noisy triangle scene is green, and the red triangle indicates where the system estimates its position.



(a) Left

(b) Right

(c) 3D primitives

(d) Pose estimation after a 1cm translation.

(e) Pose estimation after a 5cm translation.

Fig. 8. Using MoInS 3D primitives as observations. Figures (a) and (b) show a stereo view of a basket. Figure (c) shows MoInS observations. Figures (d) and (e) show the result of two pose estimation experiments; in Figure (e), visualization is rendered from the camera viewpoint whereas in Figure (d) it is rendered from a different viewpoint.

Figures 8(d) and 8(e) show preliminary results with MoInS features. A model for the basket is learned from one stereo pair (see Figure 8(c)). The model is then instantiated in a scene shot 1cm closer to the basket (Figure 8(d)) and in another scene shot 5cm closer to the basket (Figure 8(e)). The 5cm result happens to look better because it is rendered from a viewpoint similar to the stereo camera, and – as is typical for stereo reconstruction – MoInS 3D primitives are localized much more accurately in a direction perpendicular to the optical axis of the camera than in depth.

As noted above, this experiment is preliminary. For technical reasons, we were limited to translational motions along the optical axis. We plan to work on sequences involving rotations and multiple objects in the near future. The system already proved some robustness against clutter in the artificial experiments, and viewpoint-related issues will be eased by the MoInS system.

VI. Conclusion

We presented a probabilistic framework for hierarchical object representation. Hierarchies are implemented with Pairwise Markov Random Fields in which hidden nodes represent generic features, and edges model the abstraction of highly correlated features into a higher-level meta-feature. Once PMRF evidence is extracted from observations, posterior marginal pose densities for all features of the graph are inferred by the Belief Propagation algorithm.

Posterior pose densities can be used to compute a pose for a known object in an unknown scene, which we demonstrated through a series of experiments to estimate rigid body motion. We are thus able to achieve pose recovery without prior object models, and without explicit point correspondences.

Our framework is not specific to visual features and allows the natural integration of non-visual features such as haptic and proprioceptive parameters. This will potentially lead to cross-modal representations useful for robotic grasping and exploratory learning of object manipulation, which we will explore in future work.

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).

The authors thank Norbert Krüger and Nicolas Pugeault for providing MoInS data and for fruitful discussions.

References

[1] M. Burl, T. Leung, and P. Perona. Face localization via shape statistics. In *Intl. W. on Automatic Face and Gesture Recognition*, 1995.

[2] Michael C. Burl, Markus Weber, and Pietro Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume II*, pages 628–641, London, UK, 1998. Springer-Verlag.

[3] Chris Dance, Jutta Willamowski, Lixin Fan, Cedric Bray, and Gabriela Csurka. Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, 2004.

[4] I. Dhillon and S. Sra. Modeling data using directional distributions. Technical report, University of Texas, Austin, 2003.

[5] RA Fisher. Dispersion on a sphere. In *Proc. Roy. Soc. London Ser. A.*, 1953.

[6] 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.

[7] Charles F. F. Karney. Quaternions in molecular modeling. *J. Mol. Graph. Mod.*, 25, 2006.

[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] 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.

[10] Thomas K. Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.

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

[12] Justus H. Piater and Roderic A. Grupen. Toward learning visual discrimination strategies. In *CVPR*, pages 1410–1415. IEEE Computer Society, 1999.

[13] Fabien Scalzo and Justus H. Piater. Statistical learning of visual feature hierarchies. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, page 44, Washington, DC, USA, 2005. IEEE Computer Society.

[14] Fabien Scalzo and Justus H. Piater. Unsupervised learning of dense hierarchical appearance representations. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, volume 2, pages 395–398, Washington, DC, USA, August 2006. IEEE Computer Society.

[15] Erik B. Sudderth, Alexander T. Ihler, William T. Freeman, and Alan S. Willsky. Nonparametric belief propagation. *cvpr*, 01:605, 2003.

[16] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. Technical report, Mitsubishi Electric Research Laboratories, 2002.

TOWARD A VISUAL COGNITIVE SYSTEM USING ACTIVE TOP-DOWN SACCADIC CONTROL

Joyca P. W. Lacroix (jlacroix@fsw.leidenuniv.nl)
Leiden University, The Netherlands

Eric Postma (postma@micc.unimaas.nl)
Jaap van den Herik (herik@micc.unimaas.nl)
Maastricht University, The Netherlands

Jaap Murre (jaap@murre.com)
University of Amsterdam, The Netherlands

July 27, 2007

Abstract

The saccadic selection of relevant visual input for preferential processing allows for the efficient use of computational resources. Based on saccadic active human vision, we aim to develop a plausible saccade-based visual cognitive system for a humanoid robot. This paper presents two initial steps toward our objective by extending the saccade-based memory model called NIM to a plausible model of natural visual classification. As a first step, we adapt NIM to a straightforward saccade-based model for the classification of natural visual input called NIM-CLASS and evaluate the model in a face-classification experiment. As a second step we aim to approach the interactive nature of human vision by extending NIM-CLASS to NIM-CLASS^{TD} by adding active top-down saccadic control. We then assess to what extent top-down control enhances the performance on the classification task. The results show that the incorporation of top-down saccadic control benefits classification performance compared to the purely bottom-up control, reducing the amount of visual input required for correct classification. Our results lead us to the conclusion that NIM-CLASS^{TD} may provide a fruitful basis for an active visual cognitive system for a humanoid robot that allows for the efficient use of the robot's processing resources.

1 Introduction

Since the early days of computer vision, the human visual system has been an important source of inspiration for building veridical visual representations suitable for cognitive processing.[1, 2, 3, 4, 5, 6] While the computer vision methods are often based on the neurocognitive and psychophysical characteristics of the human visual processing and representational system, the saccadic nature of human vision has received much less attention. The selection of relevant visual input by means of saccades may allow for a minimization of the resources required to construct suitable representations for further cognitive processing (e.g., image recognition). Psychological studies showed that in the dynamic process of actively scanning the visual scene, saccades are guided by both bottom-up processes[7] and top-down processes. [8, 9, 10, 11, 12] Several cognitive processes combine the currently available visual input with stored knowledge and the goals and plans of the viewer to select the most relevant visual input.[10, 13]

The psychological and biological insights of saccade-based human vision provide useful guidelines for building a plausible visual cognitive system that is able to perceive and interact with the environment in an efficient and natural way. PACO-PLUS (http://www.paco-plus.org/) is an ongoing project that aims at the design of a cognitive system within a humanoid robot that is able to perceive and interact with a natural environment. One of the key objectives of the project is the development of perceptual classes of natural input. Within the context of PACO-PLUS, we aim to develop a pscyhologically and biologically plausible saccade-based visual cognitive system for a humanoid robot. As a starting point we take the recently developed Natural Input Memory model (NIM) which is a saccade-based visual memory model for the recognition of natural images.[14] NIM encompasses a biologically informed perceptual front-end that selects local samples (i.e., eye fixations) from natural images by means of saccades and translates these into feature-vector representations. These representations are used to make memory-based decisions by a computational memory back-end that is based on the mathematical psychology tradition. Although NIM's saccadic selection is based on visual saliency, we will extend the model with an original approach to top-down saccadic control that relies on cognitive systems.

This paper presents two initial steps toward the realization of a visual cognitive system for a humanoid robot capable to classify natural visual input on the basis of saccadically selected samples. As a first step, we present a saccade-based classifier of natural images called NIM-CLASS that is based on NIM. Subsequently, we aim to approach the interactive nature of natural vision by extending NIM-CLASS with an active top-down saccadic control mechanism. The extended model is called NIM-CLASS^{TD}. We then assess to what extent the use of top-down saccadic control as employed by NIM-CLASS^{TD} improves the performance on a face-classification task compared to the purely bottom-up saccadic control as employed by NIM-CLASS.

The outline of the remainder of this paper is as follows. In section 2, we present NIM-CLASS, a saccade-based model for the classification of natural visual input that is based on NIM. This is followed in section 3 by a description of the classification experiment that was used for our classification studies involving the classification of faces. In section 4, the classification performance of NIM-CLASS is evaluated on the face-classification task. Subsequently, section 5 extends NIM-CLASS to NIM-CLASS^{TD} by introducing top-down saccadic control to direct saccades toward relevant spatial locations in an image. After that, section 6 assesses the classification performance of NIM-CLASS^{TD}. In section 7, we discuss bottom-up and top-down gaze control models, examine the scalability of the NIM-CLASS variants, and provide a comparison with existing models that have been tested for classification using the same data set of stimuli. Finally, in section 8, we summarize the results and draw conclusions on the feasibility of NIM-CLASS^{TD} as a plausible visual cognitive system for a humanoid robot.

2 NIM-CLASS

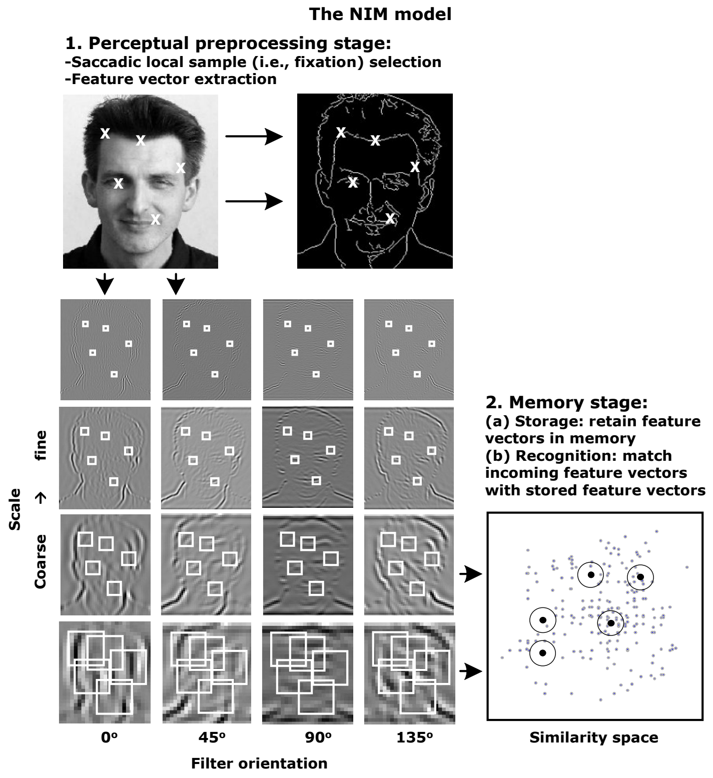
NIM-CLASS is a model for the classification of natural images. It is based on NIM that realizes a saccade-based memory model for the recognition of natural images.[14] NIM encompasses the following two stages.

1. A saccade-based perceptual preprocessing stage that selects local image samples and translates these into feature vectors.
2. A memory stage comprising two processes:
   (a) a storage process that stores feature vectors in a straightforward manner;
   (b) a recognition process that compares feature vectors of a newly presented image with previously stored feature vectors.

Fig. 1 presents a schematic overview of NIM. The left and right side of the figure correspond to the perceptual preprocessing stage (left) and the memory stage (right), respectively. Inspired by saccades in human vision, the perceptual preprocessing stage saccadically selects image samples (i.e., fixations) in a saliency-based manner (along the contours in the image). For each fixation, visual input is translated into a feature vector that resides in a similarity space. The translation is realized using a biologically informed method that involves a multi-scale wavelet decomposition (we use the steerable pyramid[15]) followed by a principal component analysis. This method from the domain of visual object recognition models the first stages of processing of information in the human visual system (i.e., retina, LGN, V1/V2, V4/LOC).[16] NIM applies the method in a saccade-based manner to build representations of fixated image parts that together constitute the feature-vector representation of an image. The memory stage stores the feature-vector representation (the storage process) and makes memory-based decisions (e.g., recognition) by matching an incoming feature-vector representation with previously stored representations. For a more detailed description we refer to [14]. While NIM is a model for recognition of natural images, here we show that it can readily be adapted into a model for classification of natural images which we call NIM-CLASS. The feasibility of adapting NIM for classification has been shown recently by [17] who combined NIM's preprocessing stage to transform fixated image parts into feature vectors with a Bayesian version of the memory stage in their NIMBLE model and successfully applied it to face classification. NIM-CLASS uses a slightly different approach that also adopts NIM's preprocessing stage, but introduces a different memory stage based on a nearest neighbor classifier that has been demonstrated to be highly suitable for object classification.[18] Below, we discuss the two processes of the NIM-CLASS memory stage: the storage process (2.1) and the classification process (2.2). The storage and classification processes correspond to the training and the testing stages that are commonly distinguished in supervised learning.[19]

2.1 The storage process

The NIM-CLASS storage process retains (i.e., stores) saccadically selected preprocessed samples of natural images (i.e., fixations) that belong to a certain class. For NIM-CLASS, each image represents an instance of a class. Therefore, in contrast to the original NIM that stores unlabeled feature vectors, NIM-CLASS stores class labels with each feature vector corresponding to the class associated with the image (i.e., '1' for class 1, '2' for for class 2, and so forth).



The NIM model
1. Perceptual preprocessing stage:
-Saccadic local sample (i.e., fixation) selection
-Feature vector extraction
Scale
Coarse → fine
2. Memory stage:
(a) Storage: retain feature vectors in memory
(b) Recognition: match incoming feature vectors with stored feature vectors
0°
45°
90°
135°
Filter orientation
Similarity space

Figure 1: The Natural Input Memory model (NIM). Reproduced from Lacroix et al. (2006)

2.2 The classification process

The NIM-CLASS classification process employs a naive Bayesian method that is based on an incremental estimate of the class-dependent probabilities.[19] In the classification process, each fixation of the test image (i.e., each test feature vector) contributes to an *n*-bin histogram, the bins of which represent the 'beliefs' in the *n* different classes. For each test feature vector, the bin that corresponds to the label of its nearest neighboring stored labeled feature vector (acquired in the storage process) is incremented (e.g., if the stored labeled feature vector that is closest to the test feature vector has label '1', bin 1 is incremented). Finally, upon the last fixation, the class with the largest bin (i.e., belief) determines the classification decision. This heuristic classification process could readily be extended into a Bayesian approach in which each fixation updates class-conditional probabilities according to the Bayes update rule.

3 Classification experiment

In our experiments, we evaluate the ability of NIM-CLASS to classify natural images of faces. Below, we discuss the classification task (3.1), the data set (3.2), and the experimental procedure (3.3).

3.1 The classification task

The classification task entails the identification of a natural image of a frontal face with variations in facial expression, illumination (location of the light source), and occlusion (sun glasses and scarf). For each individual, there are 13 views in total (see Fig. 2). Humans are generally able to identify a face after a single encounter only, despite variations in appearance.[20] Inspired by this fact, NIM-CLASS is evaluated on a task in which the training set (i.e., the study list) consists of a single image for each class and the test set (i.e., the test list) of the twelve remaining images. In this respect, our evaluation differs from most evaluations in machine learning, where the training set consists of a much larger fraction of the data set (see also section 7.3).

3.2 The data set

For the face-classification task, we chose to use the AR data set that contains over 4,000 images corresponding to the faces of 126 individuals.[21] For each individual, the AR data set includes a sequence of 13 images of frontal view faces with different facial expressions, illumination conditions, and occlusions. For the experiment, we selected the sequence of 13 images (i.e., views) of the first 10 male individuals of the AR data set as our data set. All face images were downscaled to 165 × 165 pixels. Fig. 2 shows an example of the sequence of 13 views of one individual. The first (standard) view of each individual was selected for the study list, the remaining 12 views were assigned to the test list.

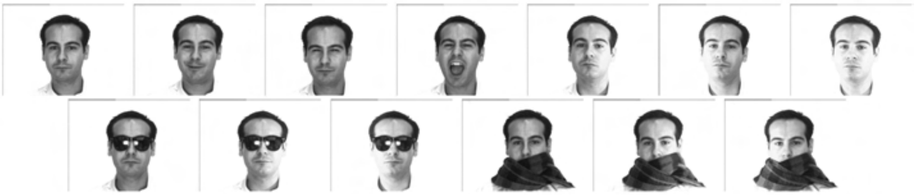


Figure 2: Example of the 13 views of one individual from the AR data set.

3.3 The experimental procedure

The face-classification experiment entailed a study and a test phase. During the study phase, we presented NIM-CLASS with the images from the study list containing the first view of each of the $n = 10$ individuals (i.e., the study faces). For each study face, NIM-CLASS extracted and stored s labeled feature vectors. Then during the test phase, the model was given the images from the test list (i.e, the 12 test faces) of each of the $n = 10$ studied individuals. For each of the test faces, the model extracted t test feature vectors to classify the face as one of the $n = 10$ individuals that it had previously encountered. To assess how the NIM-CLASS classification performance varied as a function of the number of storage fixations s and the number of test fixations t, the experiment was repeated for values of s and t in the range from 10 to 100, i.e., $s, t \in \{10, 20, ...100\}$.

4 Classification by NIM-CLASS

Below, we present the NIM-CLASS results for the face-classification task (4.1). Subsequently, we evaluate the realism of NIM-CLASS as a plausible model of natural visual classification by comparing viewing time and fixation selection by NIM-CLASS with that by humans (4.2).

4.1 Classification results

Table 1 presents the percentages of correctly classified test faces for a range of values of the number of storage fixations s and the number of test fixations t. The NIM-CLASS classification performances range from just above chance level (16%) for $s = t = 10$ to a good performance of 89.0% for $s = t = 100$. Evidently, NIM-CLASS is capable of exhibiting a good performance provided that a sufficient number of fixations is made.

The results show, not surprisingly, that the performance increases both with the number of storage fixations and the number of test fixations. Increasing the number of stored fixations s, improves the performances more than increasing the number of test fixations t. For small s values, the performance hardly increases with t. Evidently, increasing the number of test fixations is only useful when a sufficient number of feature vectors was stored previously. From a statistical perspective this makes sense. A proper approximation of the true distribution of feature vectors in a similarity space associated with a single face requires a sufficient number of samples (fixations) of that face.

Table 1: Percentages of correctly classified faces for a range of values of the number of storage fixations s and the number of test fixations t.

	t	10	20	30	40	50	60	70	80	90	100
s											
10		16.0	18.2	20.6	22.1	23.6	23.7	24.4	25.3	25.5	26.2
20		21.3	26.3	29.5	32.1	35.5	38.3	39.3	41.1	42.7	43.5
30		26.5	32.8	38.1	42.5	46.3	49.0	52.0	53.3	55.5	57.3
40		30.0	39.5	45.7	51.1	55.1	58.6	60.8	63.1	64.5	66.8
50		34.0	45.2	51.7	57.0	61.8	64.9	68.0	70.0	71.5	73.7
60		36.7	49.2	57.0	62.7	66.9	70.7	73.7	75.3	77.3	78.5
70		39.8	52.9	61.8	67.7	71.2	75.3	77.8	79.6	80.9	82.5
80		42.7	57.0	65.9	70.9	75.4	77.9	80.7	82.9	84.3	85.4
90		45.7	60.1	68.3	73.8	78.3	81.1	83.3	84.8	85.9	87.4
100		47.6	63.1	71.3	77.0	80.6	83.2	84.7	87.1	87.8	89.0

Overall, the Nim-Class classification results demonstrate that natural images of frontal faces under a variety of potentially disturbing conditions can be classified correctly using a classification process that compares (a sufficient number of) stored local image samples (i.e., fixations) acquired during one encounter (i.e., one stored view) to incoming local samples.

4.2 Comparison with humans

Considering our aim to build a visual cognitive system based on human vision, we compare the Nim-Class performance with that of human face identification in a natural setting. Below, we compare viewing time (4.2.1) and saccadic control (4.2.2) by Nim-Class with that by humans.

4.2.1 Viewing time by Nim-Class and by humans

The number of storage and test fixations extracted by Nim-Class can be interpreted as the amount of viewing time of the image during the study and test phase, respectively. Dividing the number of fixations by five provides a rough estimate of the number of seconds the image is inspected, since humans make about five fixations per second.[10, 22] As the results show, the Nim-Class performance relies heavily on the amount of viewing time during the study phase. This accords with results from several psychological studies indicating that memory for visual information increases with the amount of viewing time during the study phase.[23, 24, 25] Moreover, it is interesting that a considerable percentage of faces (say $\approx 75\%$) is classified correctly after a short viewing time of about 8 seconds (40 fixations) during the test phase, provided that there was a sufficiently long viewing time of about 20 seconds (100 fixations) during the study phase.

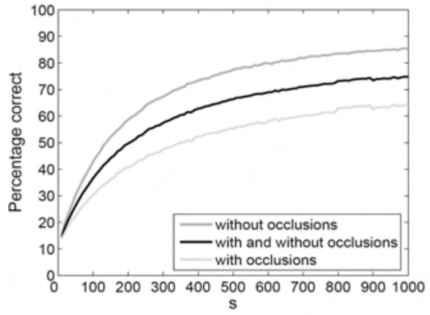


Figure 3: Percentages of correctly classified faces for a fixed number of test fixations, $t = 5$, as a function of the number of storage fixations s for the test views without occlusions (dark-grey line), with occlusions (light-grey line), and with and without occlusions (black line).

We performed additional simulations to assess in more detail to what extent NIM-CLASS is able to classify the test faces correctly on the basis of a brief viewing time during the test phase. In these simulations, the experiment was repeated for values of s that range from 10 to 1000, i.e., $s \in \{10, 20, \ldots 1000\}$ which corresponds to about 2 seconds to 200 seconds of viewing time during the study phase, and the number of test fixations was set to $t = 5$, which corresponds to approximately one second of viewing time during the test phase. Fig. 3 presents the NIM-CLASS performance for a fixed number of test fixations, $t = 5$, as a function of the number of storage fixations s. To illustrate the differences between the performances for the faces without and with occlusions, Fig. 3 shows the average performances across the six test views without occlusions (dark-gray line), the average performances across the six test views with occlusions (light-gray line), and the average performances across all the twelve test views (black line), separately. For the faces with occlusions, a degraded performance is observed compared to the performance for faces without occlusions. This is so, because the probability that a sufficient amount of relevant visual information is gathered for correct classification diminishes rapidly when one or more of the limited number of only five test fixations happen to be selected from occluded regions of the face. Still, it can be said that NIM-CLASS is able to reach a considerable average classification performance on the basis of a brief viewing time during the test phase, provided NIM-CLASS has studied the face for a sufficiently long time. The same holds for human vision, for which it is known that a brief viewing time will allow for correct identification, provided the face is sufficiently familiar to the observer.[26, 20]

4.2.2 Saccadic control by NIM-CLASS and by humans

NIM-CLASS saccadically selects image samples (i.e., fixations) on the basis of their visual saliency (along the contours). Several behavioral studies showed that in human vision bottom-up processes draw the eyes toward salient visual features such as high edge density and local contrast.[27, 7] Based on these findings, many models of gaze control employed a bottom-up approach.[28, 29, 30] Often, a so-called 'saliency map' is constructed that marks those image regions that are visually distinct from their surround in one or more visual features.[31] Then the gaze is directed to locations that are marked as highly salient on the saliency map. In fact, our contour-based saccadic control for the selection of fixations can be regarded as a realization of the bottom-up approach in which contours are the salient features. Evidently, the saccadic fixation selection of NIM-CLASS, can hardly be considered to agree with the active context-dependent scanning of a visual scene that humans perform.[32] In addition to the available visual input, human saccadic control draws on several cognitive systems.[9, 10, 11]

While bottom-up processes (based on visual saliency) have often been used to control the gaze in artificial systems, the use of top-down processes has not been examined as often. Top-down processes employ stored knowledge and the goals of the viewer to select the most relevant gaze location.[10, 13] Several studies showed that human gaze control relies more on top-down processes than on bottom-up processes when performing an active visual task with meaningful stimuli.[11] The top-down processes are driven by several cognitive systems, including: (i) short-term episodic memory for previously attended visual input,[33, 10] (ii) stored long-term knowledge about visual, spatial, and semantic characteristics of classes of items or scenes acquired through experience,[10] and (iii) the goals and plans of the viewer.[8, 34, 10] To adhere to the use of top-down processes for saccadic control in human vision, sections 5 and 6 explore the use of top-down saccadic control and investigate to what extent this may aid performance on the classification task.

5 Top-down Saccadic Control: Extending NIM-CLASS to NIM-CLASS^{TD}

Inspired by fixation selection in human vision, this section extends NIM-CLASS to NIM-CLASS^{TD} by extending it with top-down saccadic control. NIM-CLASS^{TD} employs the top-down saccadic control during the classification of the test faces (i.e., in the classification process) and adopts the bottom-up (i.e., contour-based) saccadic control of NIM-CLASS when it stores a face during the first encounter (i.e., in the storage process). Top-down saccadic control during classification is based on two main insights from psychological and neurocognitive studies about human gaze control demonstrating that: (i) saccadic local input selection is preceded and informed by a preselective holistic image processing based on global image features,[35, 26, 36, 37] and (ii) episodic knowledge about previously attended item parts provides detailed item-specific information that may contribute to the recognition or classification of the item.[33, 10, 24]

Inspired by the first insight that the human visual system computes a global summary of the entire image in a preselective initial glance used for saccadic selection,[38] NIM-CLASS^{TD} builds a global image representation by extracting global features from the entire image. The role of the global representation in the model is to create a so-called gist of the scene that is used to inform the subsequent saccadic selection of local image samples.[11, 37]

Inspired by the second insight that humans rely on stored information about previously attended visual parts for saccadic control, NIM-CLASS^{TD} implements an entropy-based mechanism[39] that uses stored episodic knowledge about attended image parts (i.e., the labeled feature vectors that were acquired during the storage process) to direct saccades to locations that are likely to contain relevant visual input to solve the classification task.

Below, we discuss the two processes of the memory stage of NIM-CLASS^{TD}: (i) the storage process (5.1), and (ii) the classification process (5.2).

5.1 The storage process

The storage process of NIM-CLASS^{TD} stores labeled feature-vectors in a similar way as the original NIM-CLASS, except that for each fixation NIM-CLASS^{TD} stores the coordinates of the fixation location. In addition to the stored labeled feature vectors, a (labeled) global image representation is stored that is used to inform the saccadic control mechanism. Several researchers have shown that a reliable global representation can be constructed by pooling together the features that are used for local receptive-field based representations (such as those implemented by the steerable pyramid transform that we use for local feature extraction) over a large image region.[40, 37] In line with this idea, NIM-CLASS^{TD} first downscales the image to a low-resolution version of 8×8 pixels and then filters the image using the same directional derivative wavelets that are also used to transform saccadically selected locals samples.[37] The filter responses are put together in a vector which is reduced to 50 dimensions using PCA.

The coordinate labels and the global image representation are used for top-down saccadic fixation selection in the classification process.

5.2 The classification process

The classification process of NIM-CLASS^{TD} involves a top-down saccadic control mechanism that selects fixations on the basis of: (i) the gist of a scene, and (ii) short-term episodic knowledge. In NIM-CLASS, the gist of a scene corresponds is based on the stored global representation and the short-term episodic knowledge corresponds to the labeled feature vectors that were acquired during the storage process directly preceding the current classification process.

For the implementation of the top-down fixation-selection mechanism, we rely on the notion of Shannon's entropy.[39] Shannon introduced entropy as a measure of uncertainty. In order to decide in the most efficient way to which class a new item belongs, a system should select new input that minimizes the entropy, i.e., the uncertainty about the class membership. In NIM-CLASS, uncertainty is represented by the histogram in which the heights of the bins represent the beliefs in the different classes. Considering the uncertainty, the top-down saccadic control mechanism selects those fixation locations that contain the most relevant information to decide upon the class of the face under consideration (i.e., that minimize the entropy or uncertainty about the class). In order to do so, the fixation-selection mechanism of the classification process in NIM-CLASS^{TD} uses the short-term episodic knowledge about attended parts of recently encountered faces (i.e., the stored labeled feature vectors).

To select the fixation locations that minimize the entropy (i.e., the locations that contain the most relevant information for classification) the mechanism proceeds as follows. First, the global representation of the test image is compared with the stored labeled global representations. The two bins that correspond to the labels of the two nearest neighboring stored global representations (acquired in the storage process) are incremented (corresponding to the preselective construction of the gist of the scene.[37] For each subsequent saccade, it first chooses the two most likely classes, P and Q, by selecting the two highest bins in the histogram. Subsequently, it selects the fixation location that best discriminates between the two classes P and Q (i.e., contains the most relevant visual input with respect to P and Q). The selection relies on:
(i) the distances between the feature vectors of the two classes; and
(ii) the distances between the spatial fixation locations from which they originated.

The idea behind the selection is that spatially adjacent fixations within one class give rise to similar feature vectors. Hence, the fixation mechanism searches for a pair of feature vectors p and q coming from classes P and Q, respectively, that originate from relatively close spatial locations and at the same time are relatively distant from each other in the representation space.

We implemented this idea heuristically. Below, we provide the steps followed by the fixation-selection mechanism:
(i) Define the two classes that have the largest belief as the target classes, P and Q.
(ii) For each possible pair of feature vectors p and q coming from target classes P and Q, respectively, calculate the ratio $d(p,q)/d((x,y)_p,(x,y)_q)$, where $d(p,q)$ is the Euclidean distance between feature vectors p and q in the representation space and $d((x,y)_p,(x,y)_q)$ is the Euclidean distance between the spatial coordinates (x,y) of p and q.
(iii) Select the two feature vectors p and q for which the ratio is the highest.
(iv) Define the target location as the contour location that is closest to the midpoint of the line connecting the spatial coordinates of p and q.
(v) Select the contour in the test image that is closest to the target location as the location to be fixated next. If this location has been fixated before, go back to step 3 and take the next highest ratio in line. Moreover, in the highly unlikely event that all locations that are selected on the basis of the ratios have been visited, select a random fixation location.

6 Classification by NIM-CLASS^{TD}

Below, we present the results for the face-classification task performed by NIM-CLASS^{TD} (6.1) and compare the classification performance of NIM-CLASS and NIM-CLASS^{TD} (6.2).

Table 2: The Nim-Class^{TD} classification performance for a range of values of the number of storage fixations s and the number of test fixations t.

	t	10	20	30	40	50	60	70	80	90	100
s											
10		48.4	47.4	46.1	45.0	44.4	43.5	41.4	40.6	40.0	39.3
20		58.0	61.0	62.8	63.6	64.6	64.0	65.2	64.5	64.1	64.7
30		60.1	66.1	68.6	70.8	72.0	73.0	74.2	74.6	75.4	75.9
40		63.1	68.5	71.8	74.1	76.6	77.7	79.3	80.1	80.9	81.9
50		64.6	71.1	74.6	77.6	79.5	80.5	82.1	83.6	84.1	84.8
60		66.5	72.6	76.7	79.2	82.2	82.6	85.0	85.8	86.4	87.2
70		68.0	75.0	79.3	81.3	83.5	85.4	86.2	87.6	88.1	89.1
80		69.1	76.8	80.2	83.0	85.1	86.7	87.9	88.6	89.3	89.8
90		70.0	77.1	81.4	84.7	86.1	87.7	88.8	89.9	90.1	91.0
100		71.4	78.6	82.6	85.5	87.4	89.1	90.0	90.5	91.4	92.2

6.1 Classification results

Table 2 presents the percentages of correctly classified test faces as a function of the number of storage (s) and test (t) fixations for Nim-Class^{TD}. The Nim-Class^{TD} classification performance reaches a performance of 92.2% for $s = t = 100$. As for the original Nim-Class, the overall results of Nim-Class^{TD} show that performance increases with the number of storage fixations s and the number of test fixations t and the performance increases more with s than with t. As was demonstrated for Nim-Class, the results of Nim-Class^{TD} show that increasing the number of test fixations t becomes useful when a sufficient number of feature vectors was stored previously. In the case of a very limited number of $s = 10$ stored fixations, an increase in the number of test fixations seems to even harm the classification performance suggesting that the system has stored too few fixations to support the intelligent selection of fixations.

6.2 Discussion and comparison of classification results

Below we review and discuss the classification performances of Nim-Class and Nim-Class^{TD}.

The results show that extending Nim-Class with top-down saccadic control to select relevant locations during classification, improves the performance on the classification task. To allow for easy comparison, Fig. 4 displays the performances of Nim-Class and of Nim-Class^{TD} in a surface plot. Evidently, Nim-Class^{TD} directs saccades to locations that are more relevant to perform the classification task than those selected by the original Nim-Class. In Nim-Class^{TD}, the saccadic control mechanism actively constructs a fixation sequence based on: (i) a preselective gist of the scene, (ii) the task to be solved (i.e., classification), and (iii) the stored episodic knowledge about previous encounters with particular faces (i.e., the stored labeled feature vectors). Thereby it acknowledges the important role that these processes are

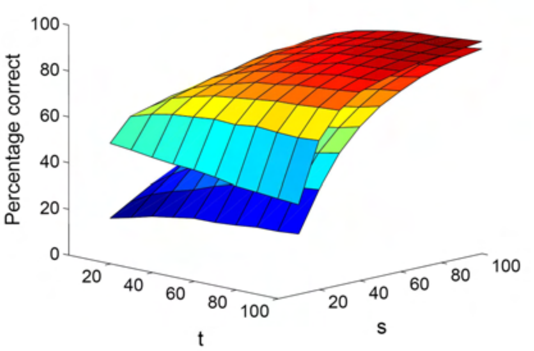


Figure 4: A comparison of the NIM-CLASS (lowest surface) and NIM-CLASS^{TD} (top surface) classification performances as a function of the number of storage fixations s and the number of test fixations t.

known to play in human gaze control.[41, 34, 10, 37] The active strategy employed by NIM-CLASS^{TD} during classification ensures that the locations are fixated that are known to discriminate well among the two most likely classes. Therefore, the model is better able to form the correct classification decision. This is particularly so, when a limited number of fixations are made. When a large number of fixations are made, a sufficient amount of relevant visual information is gathered for correct classification even when fixations are taken randomly along the contours. Using fewer fixations, the probability that a sufficient amount of relevant visual information is gathered for correct classification decreases. Therefore, performance differences between the original NIM-CLASS and the NIM-CLASS^{TD} models are most pronounced for small t values.

7 General discussion

Below, we discuss bottom-up and top-down gaze-control models (7.1). Subsequently, we discuss the scalability of the two NIM-CLASS models (7.2). Finally, we compare the NIM-CLASS and the NIM-CLASS^{TD} classification performances to the performances of existing classification models in the domain of artificial intelligence that have been tested with the AR data set that was used for our studies (7.3).

7.1 Bottom-up and top-down gaze-control models

Two types of models of saccadic control (also called gaze-control models) can be distinguished: (i) bottom-up models, and (ii) top-down models. Below, we first discuss bottom-up gaze-control models. Then, we compare top-down gaze control in NIM-CLASS^{TD} with other gaze-control models that use a top-down approach.

7.1.1 Bottom-up gaze-control models

Until now, the bottom-up or stimulus-driven approach has been the dominant approach to model gaze control. Bottom-up gaze-control models generally assume that fixation locations are selected in a bottom-up manner based on the image properties.[31, 29, 30] These models create a saliency map that marks the saliency of each image location. Saliency is defined by the distinctiveness of a region from its surround on certain visual dimensions. Since locations with a high visual saliency are assumed to be highly informative, the gaze is directed toward highly salient locations. Often, the visual dimensions that are used to generate a saliency map are similar to the visual dimensions that are known to be processed by the human visual system such as color, intensity, contrast, orientation, edge junctions, and motion.[42, 31, 43] Also, in order to discover certain important visual dimensions for generating a saliency map, a few studies analyzed which visual dimensions best distinguish fixated image regions from non-fixated image regions.[27, 7, 13]

Several studies showed that, under some conditions, fixation patterns predicted by bottom-up gaze-control models correlate well with those observed in human subjects.[43] In a study that recorded human scan paths when viewing a series of complex natural and artificial scenes, it was found that human scan paths could be predicted quite accurately by stimulus saliency which was based on color, intensity, and orientation.[43] While the bottom-up approach may be successful in predicting human fixation patterns for some tasks, it is inaccurate in predicting fixation patterns for an active task that uses meaningful stimuli.[11, 44, 13] For example, [44] showed that a saliency model performed as accurately as a random model in predicting the scan paths of human subjects during a real-world activity. In contrast, they found that a model that used only top-down (i.e., knowledge-driven) gaze control outperformed the random model. Obviously, visual saliency alone cannot account for the human fixation patterns when performing certain tasks. Similar results were found in another study where eye movements of subjects were analyzed that viewed images of real-world scenes during an active search task.[13] It was found that a visual saliency model did not predict fixation patterns any better than a random model did. Therefore, it was concluded that visual saliency does not account for eye movements during active search and that top-down (i.e., knowledge-driven) processes play the dominant role.

7.1.2 Top-down gaze-control models

Whereas bottom-up gaze-control models use visual scene characteristics, top-down gaze-control models rely on stored knowledge and task demands to select the most relevant fixation locations.[10] This paper addresses one type of top-down saccadic control relying on short-term episodic knowledge about previously attended image parts (and also on the task demands) to actively select the relevant fixation locations. This type of top-down saccadic control relates to the approach employed by probabilistic active vision models for classification.[45] Both select visual input to reduce uncertainty about the class of a test item. The main difference between top-down gaze control in the active probabilistic models and the top-down gaze control by NIM-CLASS^{TD} concerns the nature and amount of knowledge that the mechanism uses to select relevant fixations. Active probabilistic models either consider all possible fixation selections at each time step[46, 47], consider all possible fixation selections on forehand[48], or use a fixation selection policy that is acquired on the basis of an extensive training (e.g., reinforcement learning[49]) or on the basis of an evolutionary algorithm.[50] In contrast, top-down saccadic control by NIM-CLASS^{TD} relies solely on the feature vectors that were stored during one encounter with the class instance (in the storage process). A few other models have used a top-down approach to select relevant image parts for classification on the basis of limited short-term episodic knowledge.[51] The main difference with these models concerns the explicit representation of movement sequences. The models often employ a separate 'where' (motor memory) system that uses fixed eye-movement programs acquired from previously viewing the image.[51] In contrast, NIM-CLASS^{TD} actively constructs fixation patterns during classification based on the short-term episodic knowledge about previous encounters with the faces (i.e., the stored labeled feature vectors), rather than relying on the eye-movement sequence that was performed during the first encounter.

7.2 Scalability of the models

In our studies we have not examined how the NIM-CLASS and the NIM-CLASS^{TD} classification performances scale up with the number of classes. Below we offer some perspective on the aspects that relate to the scalability of the models.

In our classification task, NIM-CLASS and NIM-CLASS^{TD} deal with 130 objects (i.e., faces) coming from 10 different classes. Obviously, this limited number of objects can hardly be considered to be representative for the large number of objects that natural systems encounter in the real world. Ideally, a plausible classification or recognition model should be able to distinguish among large numbers of objects. However, since the different NIM-CLASS models store the complete encountered visual input, classification time is linear in the number of encountered objects.[52] In order to address this problem, mechanisms can be incorporated that use the representation space in an efficient way and that ensure the maintenance of an efficient representation space. In NIM-CLASS^{TD} we introduced a mechanism that operates on the representation space in an efficient way by actively using the most relevant information in the representation space. Therefore, we may assume that NIM-CLASS^{TD} is more suitable than NIM-CLASS for upscaling to a larger number of classes because it uses the representation space in an efficient manner.

Further extension may address the maintenance of an efficient representation space. For example, the storage process can be adapted in such a way that only relevant information is stored and retained. In human vision the brain continuously makes predictions about the expected visual input at the new fixation location.[53] In a similar way a new NIM-CLASS variant may make predictions on the basis of long-term knowledge and then store only the new input that deviates significantly from the expectations (i.e., the relevant or informative input). In addition, the sparseness of the representation space may be improved by forgetting stored information that is not addressed for a sufficiently long period of time. Several neurally inspired representation techniques can be used to realize a sparse and efficient representation space even for large numbers of objects, including self-organizing maps (possibly growing upon novelty), radial basis function networks, and spiking neural networks.

7.3 Comparison with existing classification models

Several other models have been applied for the classification of the faces of the AR data set. However, the existing models: (i) generally leave out the faces with occlusions that appear to be the most difficult ones for many models or (ii) are trained on more class instances than the one view that we used for training.[54, 55, 56] An example of (ii) is a study that compared the performances of a nearest-neighbor classifier operating on a representation space based on a Principal Component Analysis (PCA) and on a Linear Discriminant Analysis (LDA) of the pixel values of the entire AR images.[54] Their classification task differed from ours in the number of classes and the number of training views. While their model was presented with 50 individuals (classes) rather than the 10 individuals that we selected, training was based on a larger set of two or even 13 views[1] rather than the one view that we used. Despite their larger set of training instances per class, performances were lower than the performances that we obtained with the different NIM-CLASS models. For the experiments with two training views, the maximum average classification performances of around 60% were obtained using PCA with 80 dimensions. For the experiments with 13 training views, the maximum average classification performances of around 87% were obtained using LDA. A more recent study examined to what extent data-dependent kernels in a nearest-neighbor classifier can enhance performance on a face-classification task.[56] Their data-dependent kernel methods were tested for classification on various data sets including the AR data set. Although their classification task used all of the 126 individuals of the AR data set (compared to our 10 face classes), it was simplified in two ways compared to our classification task. First, the face views with occlusions were left out, leading to 7 face views without occlussions per class. Second, rather than training the model on the one view per class that we did, they selected five faces for training and the remaining two views were used for testing. The performances obtained with three different data-dependent kernel methods ranged from 83.1 to 94.6. For a comparison, our NIM-CLASS and NIM-CLASS^{TD} models showed an average performance for the faces without occlusions of 96.8% and 97.4%, respectively, when we used $s = t = 100$ storage and test fixations. [2]

Although most models that were tested for classification on the AR data set were trained on more than one view per class or left out the unfavorable views for testing, a few studies used the same training and testing set as we did (i.e., one view for training and the remaining 12 views for testing).[57] For example, [57] used one view of each class for training and the remaining twelve for testing, when they compared the performances of different classification algorithms on the AR data set. In their study, they introduced the use of a Non-negative Matrix Factorization (NMF[58]) in the context of classification and compared the performance of NMF with those of the widely applied

[1] In addition to the series of 13 views of each individual, there was a second series of 13 views showing the same view but taken at another point in time; when [54] used 13 training views, the 13 views of the second series were used for testing.

[2] Since Tables 1 and 2 show average performances across all test views, these values are not found in the tables.

PCA, and two influential techniques from computer vision, a feature-based technique based on Local Feature Analysis (LFA; [59]), and a Bayesian template-based technique [60]. The techniques showed average performances for the faces without occlusions (i.e., views 2 up to 7), of about 65.0% for the template-based technique, 74.0% for the PCA technique (using 150 dimensions), 85.0% for the NMF technique and 90.0% for the LFA-based technique (as mentioned in the previous paragraph our NIM-CLASS and NIM-CLASS^{TD} models exceed these performance reaching performances of 96.8% and 97.4% using $s = t = 100$ storage and test fixations). However, average performances of each of the techniques dropped substantially for the faces with occlusions, in particular for the faces with sunglasses for which the average performance was about 8.0% for the LFA-based technique, 22.0% for the PCA technique, 27.3% for the NMF technique and 32.3% for the template-based technique. For a comparison, our NIM-CLASS and NIM-CLASS^{TD} models showed average performances for the faces occluded with sunglasses of 85.5% and 89.4%, respectively, when we used $s = t = 100$ storage and test fixations. The considerable drop in the classification performance of the techniques tested by [57] for faces with sunglasses, demonstrates that the occluded parts contain important visual information for classification with these techniques. Occlusions are known to be problematic for techniques that construct global representations, such as PCA, rather than part-based representations.[58, 61] Although, the NMF technique and also the LFA technique are more part-based than the PCA and the template-based techniques, they still rely on global image characteristics to some degree. The performances of the NIM-CLASS variants that rely on discrete local samples across the images, appear to be less disrupted by an occlussion of the eyes even when testing under different lighting conditions. Therefore, it can be said that, despite occlusions, the NIM-CLASS models can make the correct classification decision on the basis of the local samples from image regions other than the occluded regions.

8 Conclusions

This paper presented two initial steps toward the realization of a plausible model of natural visual classification based on saccadic natural vision. As a first step, we presented a saccade-based classification model of natural visual input called NIM-CLASS that is based on NIM. NIM-CLASS was tested on a face-classification task involving the identification of frontal view faces with different facial expressions, illumination conditions, and occlusions. The NIM-CLASS classification results demonstrate that natural images of frontal faces with unfavorable variations in appearance can be classified correctly using a classification process that compares (a sufficient number of) stored local image samples (i.e., saccadic eye fixations) acquired during one encounter (i.e., one stored view) to incoming local samples. As a second step we attempted to approach the active saccadic nature of human vision by extending NIM-CLASS with top-down saccadic control. The extended model, called NIM-CLASS^{TD}, implemented an original approach to saccadic control relying on cognitive systems to direct saccades to locations that are likely to contain the relevant visual input to solve the task at hand. The results demonstrate that the intelligent top-down saccadic selection of relevant visual input enhanced classification performance compared to the purely bottom-up control and reduced the amount of visual input required for correct classification. Therefore, we conclude that NIM-CLASS^{TD} may provide a fruitful basis to extend into a visual cognitive system for a humanoid robot that allows for the efficient use of the robot's visual processing resources.

9 Acknowledgments

The work described in this paper was partially conducted within the EU Cognitive Systems project PACO-PLUS (FP6-2004-IST-4-027657) funded by the European Commission and partially within the Cognition Program project Events in Memory and Environment (051.02.2002) funded by the Netherlands Organization for Scientific Research (NWO).

References

[1] D. Marr, *Vision*. (W. H. Freeman, New York, 1982).

[2] I. Biederman, Human image understanding: Recent research and a theory, *Computer Vision, Graphics, and Image Understanding* **32**, 29–73 (1985).

[3] I. Biederman, Recognition-by-components: A theory of human image understanding, *Psychological Review* **94**, 115–147 (1987).

[4] M. J. Swain and D. Ballard, Color indexing, *International Journal of Computer Vision* **7**, 11–32 (1991).

[5] S. Edelman and N. Intrator, Learning as extraction of low-dimensional representations, in R. Goldstone, D. Medin, and P. Schyns, editors, *Mechanisms of perceptual learning* **36**, 353–380. (Academic press, San Diego, CA, 1997).

[6] B. W. Mel, Seemore: Combining color, shape, and texture histogramming in a neurally inspired approach to visual object recognition, *Neural Computation* **9**, 777–804 (1997).

[7] D. J. Parkhurst and E. Niebur, Scene content selected by active vision, *Spatial Vision* **16**, 125–154 (2003).

[8] A. L. Yarbus, *Eye movements and Vision*. (Plenum Press, New York, NY, 1967).

[9] K. S. Karn and M. M. HayHoe, Memory representations guide targeting eye movements in a natural task, *Visual Cognition* **7**, 673–703 (2000).

[10] J. M. Henderson, Human gaze control during real-world scene perception, *Trends in Cognitive Science* **7**, 498–504 (2003).

[11] A. Oliva, A. Torralba, M. S. Castelhano, and J. M. Henderson, Top-down control of visual attention in object detection, in Proceedings of the *IEEE International Conference on Image Processing*, (2003) pp. 253–256.

[12] M. B. Neider and G. J. Zelinski, Scene context guides eye movements during visual search, *Vision Research* **46**, 614–621 (2006).

[13] J. M. Henderson, J. R. Brockmole, M. S. Castelhano, and M. Mack, Visual saliency does not account for eye movements during search in real-world scenes, in *Eye movements: A window on mind and brain*. (Elsevier, Oxford, UK, to appear).

[14] J. P. W. Lacroix, J. M. J. Murre, E. O. Postma, and H. J. van den Herik, Modeling recognition memory using the similarity structure of natural input, *Cognitive Science* **30**, 121–145 (2006).

[15] E. P. Simoncelli and W. T. Freeman, The steerable pyramid: a flexible architecture for multi-scale derivative computation, in *Proceedings of the 2nd Annual IEEE International Conference on Image Processing*, (Washington, DC, 1995).

[16] T. J. Palmeri and I. Gauthier, Visual object understanding, *Nature Reviews Neuroscience* **5**, 291–303 (2004).

[17] L. Barrington, T. K. Marks, and G. W. Cottrell, NIMBLE: A kernel density model of saccade-based visual memory, in *Proceedings of the 29th Annual Meeting of the Cognitive Science Society (CogSci 2007)*, (2007).

[18] F. Mattern and J. Denzler, Comparison of appearance based methods for generic object recognition, *Pattern Recognition and Image Analysis* **14**, 255–261 (2004).

[19] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. (Wiley & Sons Inc., New York, NY, 2001).

[20] A. M. Burton, R. Jenkins, P. J. B. Hancock, and D. White, Robust representation for face recognition: The power of averages, *Cognitive Psychology* **51**, 256–284 (2005).

[21] A.M. Martinez and R. Benavente, The AR face database, *CVC Technical Report #24* (1998).

[22] E. McSorley and J. M. Findlay, Saccade target selection in visual search: Accuracy improves when more distractors are present, *Journal of Vision* **3**, 877–892 (2003).

[23] G. R. Loftus, Eye fixations and recognition memory for pictures, *Cognitive Psychology* **3**, 525–551 (1972).

[24] T. Mäntylä and L. Holm, Gaze control and recollective experience in face recognition, *Visual Cognition* **13**, 365–386 (2006).

[25] D. Melcher, Accumulation and persistence of memory for natural scenes, *Journal of Vision* **6**, 8–17 (2006).

[26] J. M. Findlay and I. D. Gilchrist, *Active vision: The psychology of looking and seeing*. (Oxford University Press, New York, NY, 2003).

[27] S. K. Mannan, K. H. Ruddock, and D. S. Wooding, The relationship between the locations of spatial features and those of fixations made during visual examination of briefly presented images, *Spatial Vision* **10**, 165–188 (1996).

[28] J. Braun, C. Koch, D. K. Lee, and L. Itti, Perceptual consequences of multilevel selection, in J. Braun, C. Koch, and J. L. Davis, editors, *Visual attention and cortical circuits*, 215–241. (The MIT Press, Cambridge, MA, 2001).

[29] R. P. Rao, G. J. Zelinsky, M. M. Hayhoe, and D. H. Ballard, Eye movements in iconic visual search, *Vision Research* **42**, 1447–1463 (2002).

[30] L. Zhaoping and K. A. May, Psychophysical tests of the hypothesis of a bottom-up saliency map in primary visual cortex, *PLoS Computational Biology* **3**, 616–633 (2007).

[31] L. Itti and C. Koch, A saliency-based search mechanism for overt and covert shifts of visual attention, *Vision Research* **40**, 1489–1506 (2000).

[32] U. Rajashekar, L. K. Cormack, and A. C. Bovik, Visual search: Structure from noise, in *Proceedings of the Eye Tracking Research & Applications Symposium 2002*, 119–123 (New Orleans, LA, 2002).

[33] M. M. Chun, Contextual cueing of visual attention, *Trends in Cognitive Sciences* **4**, 170–178 (2000).

[34] M. F. Land and M. Hayhoe, In what ways do eye movements contribute to everday activities?, *Vision Research* **41**, 3559–3565 (2001).

[35] D. Navon, Forest before trees: the precedence of global features in visual perception, *Cognitive Psychology* **9**, 353–383 (1977).

[36] A. Oliva and A. Torralba, Building the gist of a scene: the role of global image features in recogniiton, *Progress in Brain Research: Special Issue on Visual Perception* **155**, 23–36 (2006).

[37] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson, Contextual guidance of eye movements and attention in real-world scenes: The role of global features on object search, *Psychological Review* **113**, 766–786 (2006).

[38] S. C. Chong and A. Treisman, Representation of statistical properties, *Vision Research* **43**, 393–404 (2003).

[39] C. E. Shannon, A mathematical theory of communication, *The Bell System Technical Journal* **27**, 379–423, 623–656 (1948).

[40] A. Oliva and A. Torralba, Modeling the shape of the scene: A holistic representation of the spatial envelope, *International Journal of Computer Vision* **42**, 145–175 (2001).

[41] J. M. Henderson, Effects of semantic consistency on eye movements during scene viewing, *Journal of Experimental Psychology: Human Perception and Performance* **25**, 210–228 (1999).

[42] C. Koch and S. Ullman, Shifts in selective visual attention: Towards the underlying neural circuitry, *Human Neurobiology* **4**, 219–227 (1985).

[43] D. J. Parkhurst, K. Law, and E. Niebur, Modeling the role of salience in the allocation of overt visual attention, *Vision Research* **42**, 107–123 (2002).

[44] K. A. Turano, D. R. Geruschat, and F. H. Baker, Oculomotor strategies for the direction of gaze tested with a real-world activity, *Vision Research* **43**, 333–346 (2003).

[45] G. de Croon, I. G. Sprinkhuizen-Kuyper, and E. O. Postma, Comparing active vision models, Technical Report 06-02, MICC-IKAT, Universiteit Maastricht, 2006.

[46] H. Borotschnig, L. Paletta, M. Prantl, and A. Pinz, A comparison of probabilistic, possibilistic, and evidence theoretic fusion schemes for active object recognition, *Computing* **62**, 293–319 (1999).

[47] J. Denzler and C. M. Brown, Information theoretic sensor data selection for active object recognition and state estimation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**, 145–157 (2002).

[48] T. Arbel and F. P. Ferrie, Entropy-based gaze planning, *Image and Vision Computing* **19**, 779–786 (2006).

[49] L. Paletta, M. Prantl, and A. Pinz, Reinforcement learning for autonomous three-dimensional object recognition, in *Proceedings of the 6th Symposium on Intelligent Robotics Systems*, 63–72 (Edinburgh, UK, 1998).

[50] G. de Croon, E. O. Postma, and H. J. van den Herik, A situated model for sensory-motor coordination in gaze control, *Pattern Recognition Letters: Special Issue on Evolutionary Computer Vision and Image Understanding* **27**, 287–314 (2006). Guest Editor G. Olague.

[51] I. A. Rybak, V. I. Gusakova, A. V. Golovan, L. N. Podladchikova, and N. A. Shevtsova, A model of attention-guided visual perception and recognition, *Vision Research* **38**, 2387–2400 (1998).

[52] F. Bajramovic, F. Mattern, N. Butko, and J. Denzler, A comparison of nearest neighbor search algorithms for generic object recognition, in *Proceedings of the Advanced Concepts for Intelligent Vision Systems (ACIVS 2006)*, 1186–1197 (2006).

[53] J. Hawkins and S. Blakeslee, *On intelligence*, (Times Books, New York, NY, 2004).

[54] A. M. Martinez and A. C. Kak, Pca versus lda, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**, 228–233 (2001).

[55] X. Lu, Y. Wang, and A. K. Jain, Combining classifiers for face recognition, in *Proceedings of the International Conference on Multimedia and Expo (ICME)* **3**, 13–16 (July 2003).

[56] J. Wang, J. T. Kwok, H. C. Shen, and L. Quan, Data-dependent kernels for high-dimensional data classification, in *Proceedings of the International Joint Conference on Neural Networks*, 102–107, (Montreal, Canada, 2005).

[57] D. Guillamet and J. Vitri, Non-negative matrix factorization for face recognition, *Lecture Notes in Computer Science* **2504**, 336–344 (2002).

[58] D. Lee and H. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature* **401** 788–791 (1999).

[59] P. S. Penev and J. J. Atick, Local feature analysis: A general statistical theory for object representation, *Network: Computation in Neural Systems* **7**, 477–500 (1996).

[60] B. Moghaddam and A. P. Pentland, Probabilistic visual learning for object representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**, 696–710 (1997).

[61] L. Fei-Fei and P. Perona, A bayesian hierarchical model for learning natural scene categories, *IEEE Computer Vision and Pattern Recognition*, 524–531 (2005).

Minimum Volume Bounding Box Decomposition for Robot Grasping

Royal Institute of Technology, Stockholm, Sweden
Computational Vision and Active Perception Laboratory
S-100 44 Stockholm, Sweden

Abstract— **Thinking about intelligent robots involves consideration of how such systems can be enabled to perceive, interpret and act in arbitrary and dynamic environments. While sensor perception and model interpretation focus on the robot's internal representation of the world rather passively, robot grasping capabilites are needed to actively execute tasks, modify scenarios and thereby reach versatile goals. These capabilities should also include the generation of stable grasps to safely handle even objects unknown to the robot. We believe that the key to this ability is not to select a good grasp depending on the identification of an object (e.g. as a cup), but on its shape (e.g. as a composition of shape primitives). In this paper, we envelop given 3D data points into primitive box shapes by a fit-and-split algorithm that is based on an efficient Minimum Volume Bounding Box implementation. Though box shapes are not able to approximate arbitrary data in a precise manner, they give efficient clues for planning grasps on arbitrary objects. We present the algorithm and experiments using the 3D grasping simulator *GraspIt!* [1].**

I. Introduction

In the service robot domain, researchers and programmers provide each robot with manifold tasks to fulfill in order to aid and support, e.g. clearing a table or fill a dishwasher after lunch. The knowledge about such aims might be either hard-coded or techiques applied that allow learning in a more intelligent manner, e.g. a person teaching the robot how to clear a table. Such scenarios are known as Learning- or Programming-by-Demonstration applications. However, whether in an office, in health care or in a domestic scenario, a robot has to finally operate independently to satisfy various claims, thus the handling of objects is a central issue of many service robot systems. Robot grasping capabilites are therefore essential to actively execute tasks, modify scenarios and thereby reach versatile goals in an autonomous manner.

For grasping, numerous approaches and concepts have been developed over the last decades. Designing grasping systems and planning grasps is difficult due to the large search space resulting from all possible hand configurations, grasp types, and object properties that occur in regular environments. Early work on contact-level grasp synthesis focused mainly on finding a fixed number of contact locations without regarding hand geometry [2]. Considering specifically object manipulation tasks, the work on automatic grasp synthesis and planning is of significant relevance [3], [4], [5]. The main issue here is the automatic generation of stable grasps assuming that the model of the hand is known and that certain assumptions about the object (e.g. shape, pose) can be made. Taking into account both the hand kinematics as well as some a-priori knowledge about the feasible grasps has been acknowledged as a more flexible and natural approach towards automatic grasp planning [4]. It is obvious that knowledge about the object shape, as also the task on hand, is quite meaningful for grasp planning [6].

This is important for our work, as we aim at providing a robot actuator system with a set of primitive actions, like *pick-up*, *push* or *erect* an arbitrary object on a table. For performing such basic actions, an object has to be modeled from 3D sensory input, e.g. from range or dense stereo data - but up to which detail? After giving references to related work in the following section, we motivate our approach on approximating object shapes by Minimum Volume Bounding Box (MVBB) sets in Section III. We present the algorithm in Section IV and demonstrate experiments in Section V. Section VI shows experiments on using the approach for a simulated grasping evaluation, before we conclude our work in Section VII.

II. Related Work

Modeling range data is a crucial, but also difficult task for robot grasping. The source data offered by range sensors or dense stereo camera systems is a more or less distorted and scattered cloud of 3D points of the scenario. A higher-level representation of these points as a set of shape primitives (e.g. planes, spheres or cylinders) thus gives more valuable clues for object recognition and grasping by compressing information to their core. Most approaches that consider this problem are likewise bottom-up, starting from point-clouds and synthesizing object shapes by using superquadrics (SQs). Superquadrics are parametrizable models that offer a large variety of different shapes. In the problem of 3D volume approximation stated here, only superellipsoids are used out of the group of SQs, as these are the only ones representing closed shapes. There is a multitude of state-of-the-art approaches based on parametrized superellipsoids for modeling 3D range data with shape primitives [7], [8], [9], [10], [11].

If we assume that an arbitrary point cloud has to be approximated, one SQ is obviously not enough for most objects, e.g. a screw or an office chair (see Fig. 1). The more complex the shape is, the more SQs have to be used to conveniently represent its different parts. Just for such cases, good generality is not possible using SQs with few parameters [8]. Besides the advantages of immense parametrization capabilities with at least 11 parameters, intensive research on SQs has also yielded disadvantages in two common strategies for SQ approximation. The first strategy is region-growing, starting with a set of hypotheses, the *seeds*, and let these adapt to the point set. However, this approach has not proved to be effective [9] and suffers from refinement problem of the seeds [11]. The second strategy uses a split-and-merge technique splitting up an overall shape and merging parts again, which is more adapted to unorganized and irregular data [9].

Independent of the strategy used, the models and seeds, respectively, have to be fitted to the 3D data. This is usually done by least square minimization of an inside-outside fitting function, as there is no analytical method to compute the distance between a point and a superquadric [10]. Thus, SQs are though a good trade-off between flexibility and computational simplicity, but sensitive to noise and outliers that will cause imperfect approximations. This is an important issue, as our work is based on dense stereo data, which results in more distorted and incomplete data in contrast to data points provided by range scanners which are mainly applied in related work.

III. Motivation

We observed that modeling 3D data by shape primitives is a valuable step for object representation. Sets of such primitives can be used to describe instances of the same object classes, e.g. cups or tables. However, it is not our aim to focus on such high-level classifications or identification of objects, but on grasping. We moreover approach a deeper understanding of objects by interaction instead of observation for that purpose, e.g., if there is an object that can be picked up, pushed and filled, it can be used as a cup. Processing an enormous number of data points takes time, both in approaches that use raw points for grasp hypotheses or those that try to approximate them as good as possible by shape primitives. Thus the question remains how rudimentary a model of a thing can be in order to be handled successfully and efficiently. While comparable work uses pairs of primitive feature points, e.g. [12], or a-priori known models for each object [13], we are interested in looking into which primitive shape representations might be sufficient for the task of grasping arbitrary, unseen objects.

We believe that a mid-level solution is a promising trade-off between good approximation and efficiency for this purpose. Complex shapes are difficult to process, while simple ones will give worse approximation. However, we can access valuable methods to handle approximation inaccuracies for grasping like haptic feedback, visual servoing and advanced grasp controllers for online correction of grasps. We prefer general fast online techniques instead of pre-learned offline examples, thus the algorithm's efficiency is the more important. Unknown objects are hardly parametrizable but need real-time application for robot grasping. A computation in terms of minutes for a superquadric approximation is therefore not feasible.

We adopt these motivations to propose an algorithm based on boxes as a mid-level representation. In our approach, we combine different incentives on simplicity of boxes, efficiency of hierarchies and fit-and-split algorithms:

1) We aim for *simplicity* stating the question if humans approach an apple for grasping with their hand in another way as they approach a cup, or a pen in another way as a fork? While there are surely differences in fine grasping and task dependencies, differences in approaching these objects seem quite marginal.
2) The computational efficiency of *hierarchies* has been pointed out in several other approaches that compose models with use of superquadric primitives [8], [10], [14].
3) While seed growing as a bottom-up strategy has several drawbacks, and a split-and-merge strategy both needs top-down (split) and bottom-up (merge), *fit-and-split* algorithms is purely top-down and thereby iteratively implementable in a one-way hierarchical manner.

IV. Algorithm

Following the first incentive, we chose a box representation, as boxes are very simple and roughly approximating. We base our algorithm on the minimum volume bounding box computation proposed by Barequet and Har-Peled [15]. Given a set of n 3D points, the implementation of the algorithm computes their Minimum Volume Bounding Box (MVBB) in $O(n \log n + n/\varepsilon^3)$ time, where ε is a factor of approximation. The algorithm is quite efficient and parametrizable by sample and grid optimizations [15]. Performing the computation on an arbitrary point cloud, we get one tight-fitting, oriented MVBB enclosing the data points (see the example in Fig. 2).

Our aim is now to iteratively split the box and the data points, respectively, such that the new point sets yield a better box approximation of the shape.

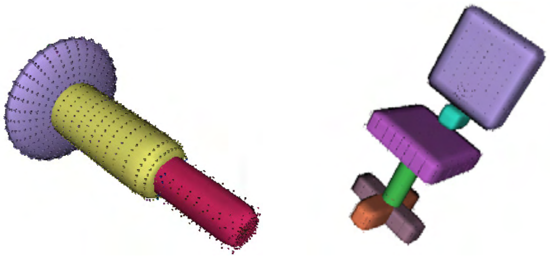


Fig. 1. Examples of range data approximated by sets of superquadrics [9].

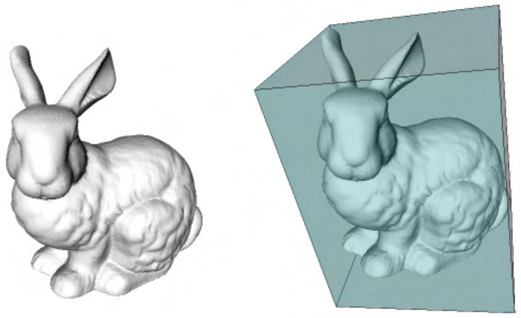


Fig. 2. The Stanford bunny model and the root MVBB of its vertices.

A. MVBB splitting

Iterative splitting of a root box corresponds to the build-up of a hierarchy of boxes. Gottschalk *et al.* [16] present the OBBTree (Oriented Bounding Box Tree) for this purpose. The goal is to efficiently detect collisions between polygonal objects by the OBBTree representation. The realization of the splitting step is quite straightforward: each box is cut at the mean point of the vertices, perpendicular to the longest axis. This is done iteratively, until a box is not dividable any more.

In our case, these strategies are quite suboptimal. We want to conveniently approximate a shape with as few boxes as possible, thus a splitting into as many small boxes as possible is against our overall aim, if we refrain from merging them again. Additionally, though the MVBB algorithm is efficient, a fitting step after each splitting consumes valuable computation time. On the other hand, splitting at the mean point is then not optimal. A heuristic to find a "good" split is needed.

Therefore, we will have to define what a "good" split is. See the bunny cuts in Fig. 3. Fig. 3(a) shows a mean cut, similar to the ones used by Gottschalk's algorithm. It is obvious that this one is not optimal for oour task, as it does not improve the approximation by the boxes of both new halfs. In addition, this split is not intuitive, meaning that it does not divide the bunny in semantic parts, e.g. head and body, as is shown in Fig. 3(b). However, such a semantic division is hard to find. Due to efficiency, we yet restricted to plane cuts instead of using non-linear cutting in the example. But even with planes, finding the best intuitive cut would correspond to an extensive search and comparison of a lot of planes, differing in position and orientation. Therefore, we decide to test only those planes parallel to the parent MVBB.

As a measure of a good split, we consult the relation of the box volume before and after performing the split. A split of the parent box is the better, the less volume the two child MVBBs include. Intuitively, this is clear, as shape approximation is better with highly tight-fitting boxes. We propose the following efficient algorithm to find the best split:

B. Best Split Computation

As discussed, we just test planes parallel to the box surfaces for the best splitting plane. Each MVBB has six sides, whereof opposing pairs are parallel and symmetric. Inbetween each of these pairs, we can shift a cutting plane. Fig. 3(d) depicts this restriction on a splitting parallel to $\overline{A}$, shifted by a distance a, and $\overline{B}$ by b and $\overline{C}$ by c, respectively. A computation of new MVBBs for each value of the *split parameters* a, b and c would take a lot of computational effort. Therefore, we estimate the best cut by first projecting the data on 2D grids which correspond to the surfaces $\overline{A}$, $\overline{B}$ and $\overline{C}$. The bunny sample data projection onto the three surface grids of the root MVBB (Fig. 2) are shown in Fig. 4. By this projection, the problem of splitting a 3D box by a surface-parallel plane has been reduced to splitting a 2D box by an edge-parallel line. For the sake of efficiency, it is thereby abstracted from the real 3D volume of the shape. The figure shows that there are six valid split directions left, two for each of the surfaces $\overline{A}$, $\overline{B}$ and $\overline{C}$.

As mentioned above, we define the best split as the one that minimizes the summed volume of the two partitions. Thus, we now test each discretized grid split along the six axes, using the split parameters. We define a split measure $\theta(\overline{P}, \overline{p}, i)$ with $\overline{P} \in \{\overline{A}, \overline{B}, \overline{C}\}$ being the projection plane to split, $\overline{p}$ being one of the two axes that span $\overline{P}$, and i as the grid value on this axis that defines the current split. Consequently, we have six possible split measures

$$\begin{aligned} &\theta_1(\overline{A}, \overline{c}, i_1),\ i_1 \in \mathbb{N}^{<c_{\max}}, \theta_2(\overline{A}, \overline{b}, i_2),\ i_2 \in \mathbb{N}^{<b_{\max}}, \\ &\theta_3(\overline{C}, \overline{a}, i_3),\ i_3 \in \mathbb{N}^{<a_{\max}}, \theta_4(\overline{C}, \overline{b}, i_4),\ i_4 \in \mathbb{N}^{<b_{\max}}, \\ &\theta_5(\overline{B}, \overline{a}, i_5),\ i_5 \in \mathbb{N}^{<a_{\max}}, \theta_6(\overline{B}, \overline{c}, i_6),\ i_6 \in \mathbb{N}^{<c_{\max}} \end{aligned} \tag{1}$$

to compare. Their minimum gives reason to the best split.

The minimization of a $\theta(\overline{P}, \overline{p}, i)$ is implemented as follows. For each i that cuts $\overline{P}$ perpendicular to $\overline{p}$ in two rectangular shapes, we compute the two resulting minimal volumes by lower and upper bounds. The i that yields the minimum value is the best cut of $\theta(\overline{P}, \overline{p}, i)$.

We define $\theta(\overline{P}, \overline{p}, i)$ as the fraction between the whole projection rectangular and the sum of the two best cut rectangles. Though this is a very approximative method, it is quite fast,

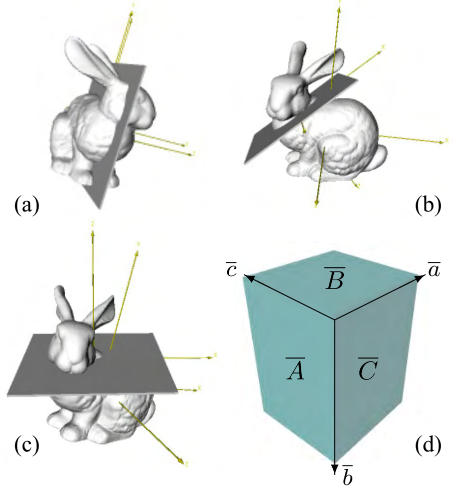


Fig. 3. Exemplary cuts of the bunny: (a) a mean cut, (b) an intuitively best cut and (c) a good cut parallel to one of the root MVBB planes. (d) Restriction to surface-parallel cutting planes in our approach.

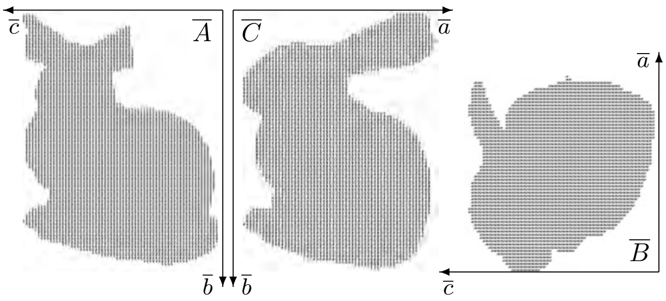


Fig. 4. The bunny sample projections onto the three surfaces of the root MVBB (Fig. 2). Note the correspondences to the surface-parallel cutting scheme in Fig. 3(d).

as rectangle volume and bound computation are simple to perform. Fig. 5 shows the best cuts for which rectangular volume and the corresponding values $\theta_{1\ldots6}$ are minimal.

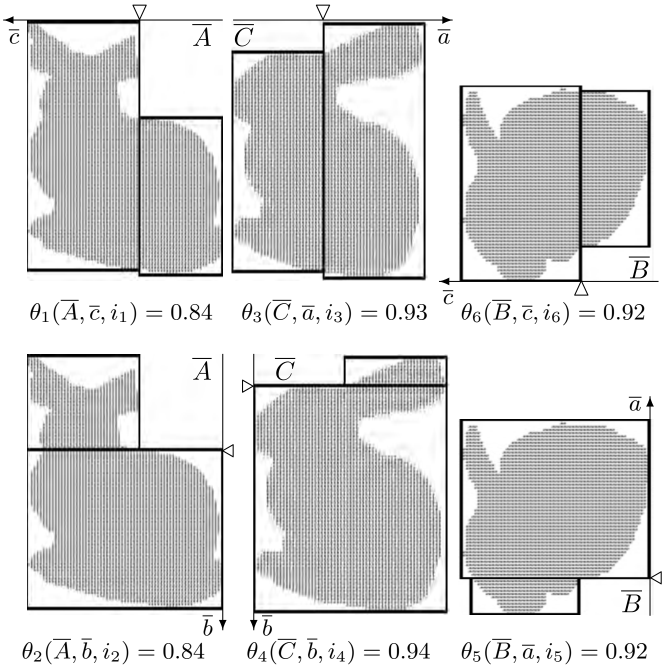
C. Fit-and-Split Hierarchy Building

According to the best split θ^*, which would be θ_1 or θ_2 in our exemplary case, the original point cloud can be divided into two subsets of the data points. These can be used as inputs to the MVBB algorithm again and are produce two child MVBBs of the root MVBB. In this way, the complete technique of fit-and-split can iteratively be performed. It is important to notice that by MVBB computation, the MVBBs will greatly differ from box cuts as depicted in Fig. 5 in orientation and scale.

Additionally, the previous step of cutting along one of the six directions is just equal to computing an approximative gain value, for the purpose of efficiency. As an iteration breaking criterion, we now subsequently test the real MVBB volume gain Θ^* of the resulting best split measure θ^*. Therefore, we compute the gain in volume defining

$$\Theta^* = \frac{V(C_1) + V(C_2) + V(A^{\setminus P})}{V(P) + V(A^{\setminus P})}, \qquad (2)$$

where A is the overall set of boxes in the current hierarchy, P is the current (parent) box, C_1, C_2 are the two child boxes that might be produced by the split, and V being a volume function. We decide further process on two constraints. First, if gain is too low, a split is not valuable. For this purpose, we include a threshold value t that can also be used as a parameter. The precision of the whole approximation can be parametrized by simply preventing a split if Θ^* exceeds t. A threshold between $t = 0.90$ and $t = 0.95$ has given good results in most of our experiments.



$\theta_1(\overline{A}, \overline{c}, i_1) = 0.84$ $\theta_3(\overline{C}, \overline{a}, i_3) = 0.93$ $\theta_6(\overline{B}, \overline{c}, i_6) = 0.92$

$\theta_2(\overline{A}, \overline{b}, i_2) = 0.84$ $\theta_4(\overline{C}, \overline{b}, i_4) = 0.94$ $\theta_5(\overline{B}, \overline{a}, i_5) = 0.92$

Fig. 5. The six best cuts along the six main box directions. The cut positions i are marked by the triangles. Below each cut, the corresponding volume value $\theta(\overline{P}, \overline{p}, i)$ is presented.

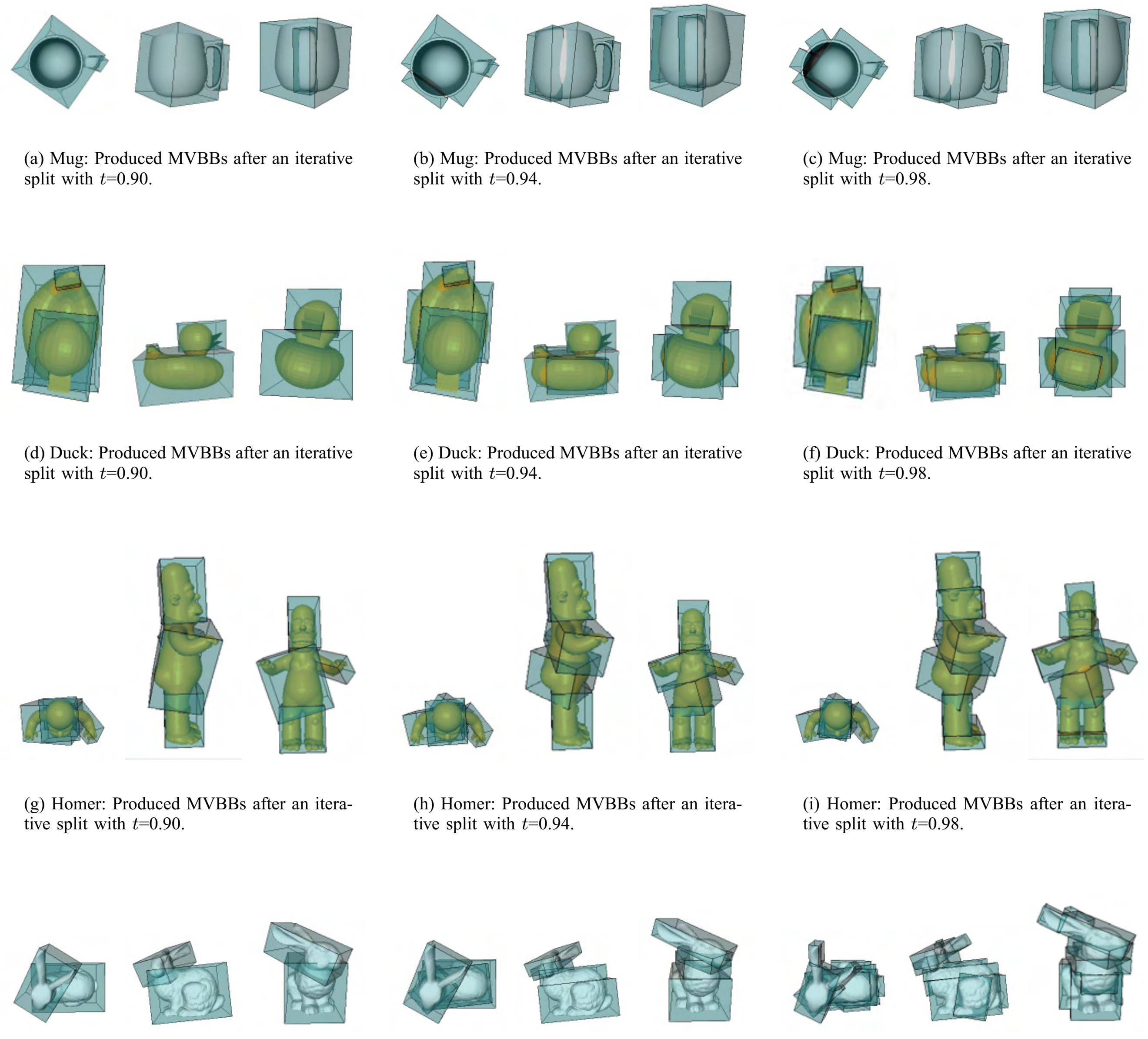
Second, we do not preserve boxes into the hierarchy that include a very low number of points. By this process, noise in the point data can be handled. It might also be important in this context that in Fig. 5 (θ_3) it would intuitively be a probably valuable next cut below the bunny's ear in the right box. However, the best split computation presented (Section IV-B) will not find this cut. Finding this cut is not that simple, especially when distorted, sparse and insecure data is provided. An add-on for the solution of this problem would therefore be more complex and time-consuming. The bunny is a very ideal model, as it is artificial, complete, and data points are very dense. As it is our aim to evaluate our algorithm also on real sensory data, we can not assume such ideal conditions. Due to these problems, we do not handle such situations, but present ideas to solve them by a different approach in the conclusions.

V. Box Splitting Experiments

We now present some experiments for the proposed fit-and-split algorithm. For all experiments, we fix the two original MVBB approximation parameters (see [15]). The grid parameter defines orientations that induce bounding box approximation in an exhaustive way, so we keep it small at 3. We decide to sample sets of 200 points, so even very large point clouds are reduced and efficiently handled. We found that these settings provide a good trade-off between quality and efficiency of each split for our application. As mentioned, we additionally disregard boxes that include too few points. For the experiments on models and sparse scans, we set this threshold to 5 points. As those point clouds are not very noisy, we can afford such a small value. Concluding, the main paramater that we are going to change in each experiments is the gain threshold t.

As we are also interested in the robustness of the algorithm with regard to different degrees of noise and data density, we evaluate the behaviour of the algorithm on several types of input data. In our 1st group, we have ideal point clouds emerging from complete and unnoisy simulative vertice models. The 2nd group consists of real laser scan excerpts. These single object data has been segmented manually from scanned scenes. The data of each object is therefore incomplete and noisy, but at least regular due to the scan sampling. The 3rd group is the most challenging of all. The data points are produced from a stereo vision system that offers three-dimensional range data by disparity. Matching of image features and disparity calculation are erroneous. Therefore, the resulting data is quite incomplete, noisy and irregular. To cover these irregularities, we increase the point threshold to at least 20 points in an MVBB and the sampling rate to sets of 300 points.

Fig. 6 shows four sample models for the 1st group, each divided with gain thresholds $t \in \{0.90, 0.94, 0.98\}$. An overview on point sets, computation time and number of boxes for the 1st group is given in Tab. I. Some samples for the 2nd group



(a) Mug: Produced MVBBs after an iterative split with t=0.90.

(b) Mug: Produced MVBBs after an iterative split with t=0.94.

(c) Mug: Produced MVBBs after an iterative split with t=0.98.

(d) Duck: Produced MVBBs after an iterative split with t=0.90.

(e) Duck: Produced MVBBs after an iterative split with t=0.94.

(f) Duck: Produced MVBBs after an iterative split with t=0.98.

(g) Homer: Produced MVBBs after an iterative split with t=0.90.

(h) Homer: Produced MVBBs after an iterative split with t=0.94.

(i) Homer: Produced MVBBs after an iterative split with t=0.98.

(j) Bunny: Produced MVBBs after an iterative split with t=0.90.

(k) Bunny: Produced MVBBs after an iterative split with t=0.94.

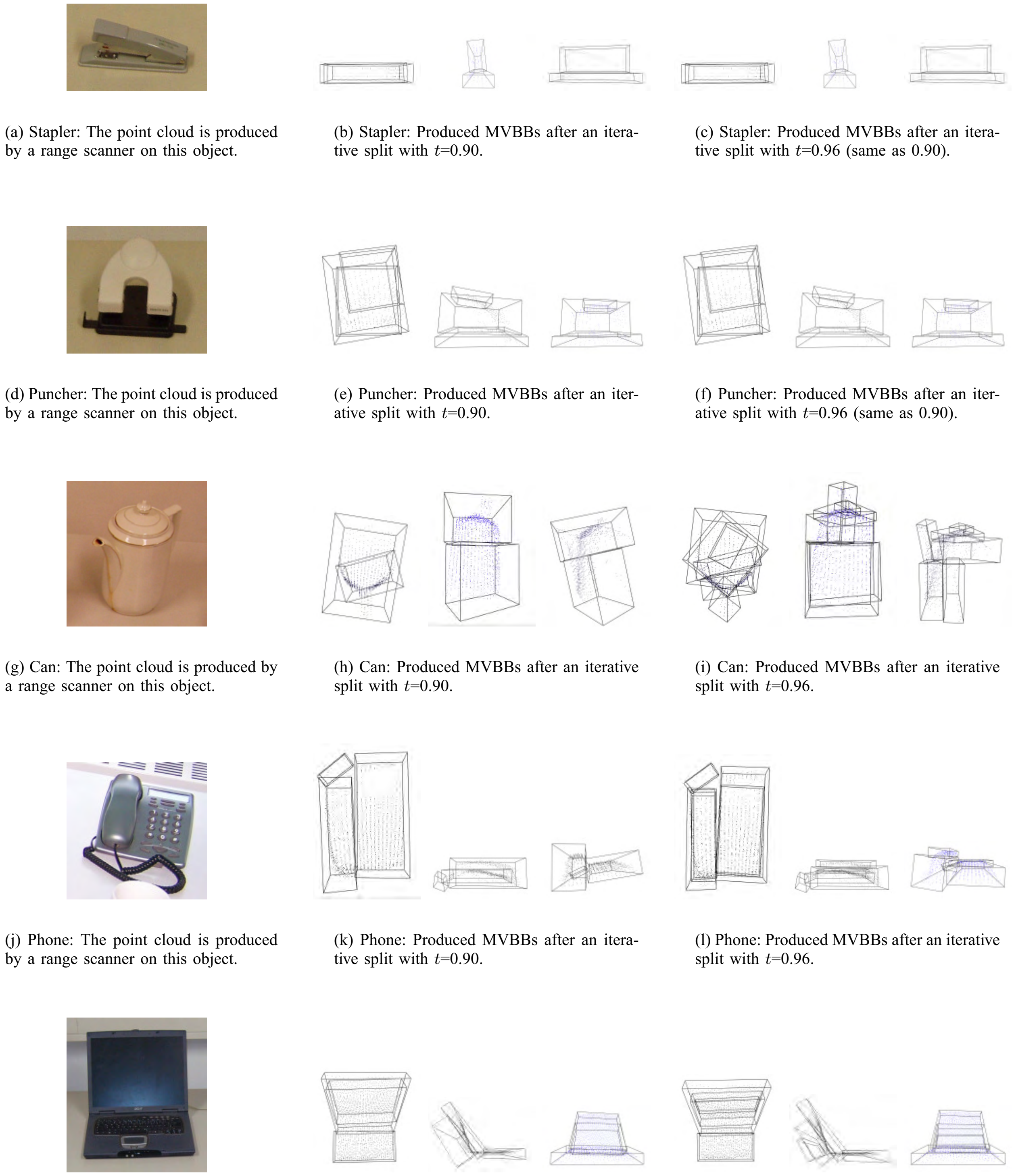
(l) Bunny: Produced MVBBs after an iterative split with t=0.98

Fig. 6. Four complete, dense and unnoisy models: Mug, Duck, Homer and Bunny. Each is fit-and-splitted according to the proposed algorithm with three different gain thresholds t (0.90, 0.94, 0.98) to show the influence of t on the algorithm.

Model	# points	time [sec] ($t = 0.90$)	# boxes ($t = 0.90$)	time [sec] ($t = 0.94$)	# boxes ($t = 0.94$)	time [sec] ($t = 0.98$)	# boxes ($t = 0.98$)
Mug	1725	4	2	6	3	8	5
Duck	1824	6	3	9	5	12	8
Homer	5103	10	4	12	5	16	8
Bunny	35947	5	2	11	4	30	11

TABLE I

Corresponding table for the experiments presented in Fig. 6.



(a) Stapler: The point cloud is produced by a range scanner on this object.

(b) Stapler: Produced MVBBs after an iterative split with t=0.90.

(c) Stapler: Produced MVBBs after an iterative split with t=0.96 (same as 0.90).

(d) Puncher: The point cloud is produced by a range scanner on this object.

(e) Puncher: Produced MVBBs after an iterative split with t=0.90.

(f) Puncher: Produced MVBBs after an iterative split with t=0.96 (same as 0.90).

(g) Can: The point cloud is produced by a range scanner on this object.

(h) Can: Produced MVBBs after an iterative split with t=0.90.

(i) Can: Produced MVBBs after an iterative split with t=0.96.

(j) Phone: The point cloud is produced by a range scanner on this object.

(k) Phone: Produced MVBBs after an iterative split with t=0.90.

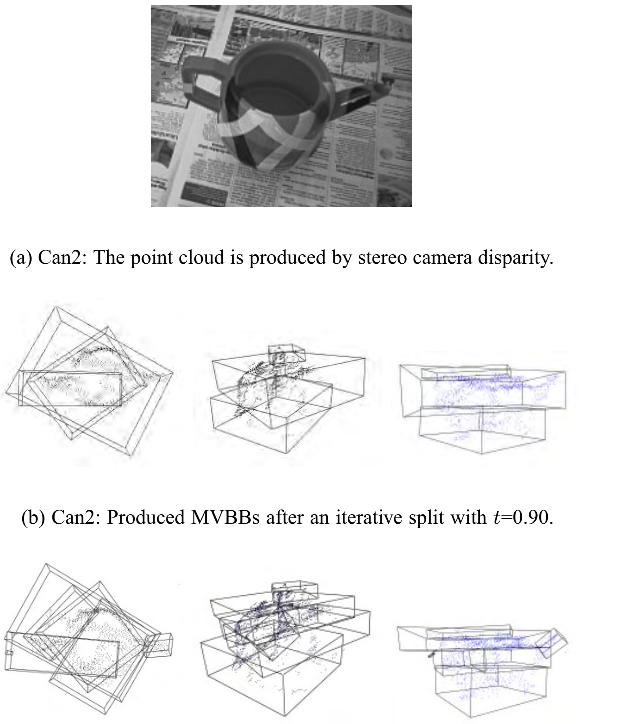
(l) Phone: Produced MVBBs after an iterative split with t=0.96.

(m) Notebook: The point cloud is produced by a range scanner on this object.

(n) Notebook: Produced MVBBs after an iterative split with t=0.90.

(o) Notebook: Produced MVBBs after an iterative split with t=0.96.

Fig. 7. Five incomplete, dense and less noisy, manually pre-segmented range scans: Stapler, Puncher, Can, Phone and Notebook. Each is fit-and-splitted according to the proposed algorithm with two different gain thresholds t (0.90, 0.96) to show the influence of t on the algorithm.



(a) Can2: The point cloud is produced by stereo camera disparity.

(b) Can2: Produced MVBBs after an iterative split with t=0.90.

(c) Can2: Produced MVBBs after an iterative split with t=0.96.

Fig. 8. An incomplete, sparse and noisy, automatically pre-segmented disparity point cloud: Can2. It is fit-and-splitted according to the proposed algorithm with two different gain thresholds t (0.90, 0.96) to show the influence of t on the algorithm.

are presented in Fig. 7, two of the 3$^{\text{rd}}$ group in Fig. 8. Statistics of these are depicted in Tab. II.

VI. Box Grasping Algorithm

A. Using GraspIt!

The best way to find a good grasp is said to be grasp candidate simulation [4], [10]. Miller *et al*. have simulated pre-models and shape primitives using their public grasp simulation environment *GraspIt!* [4]. We also base our evaluation on model-based grasping on *GraspIt!*.

For the evaluation, we create lots of worlds, each of which contains a model of the Barrett hand [17] mounted on a freely movable "Euler" robot, as the hand is not able to move itself in free space. Aditionally, an object that is to be grasped is included into the world, which is the only difference between the world files (one for each object).

Model	# pts.	time [sec] (t=0.90)	# boxes (t=0.90)	time [sec] (t=0.96)	# boxes (t=0.96)
Stapler	313	2	2	2	2
Puncher	449	3	3	3	3
Can	1266	4	2	10	6
Phone	1461	5	3	8	5
Notebook	4199	6	3	8	4
Can2	9039	10	3	24	7

TABLE II
Corresponding to the experiments shown in Fig. 7 and Fig. 8.

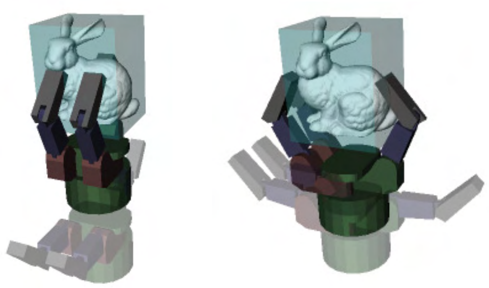


Fig. 9. Each singular box can be grasped in twelve different simple ways: it has six faces, each defined by two perpendicular directions. The picture shows one of these pairs during the grasp approach (shadow) and during the final grasp.

After an experiment is initialized, the first iteration of the MVBB algorithm is performed as proposed in Section IV-C. All levels of generated boxes are subsequently inserted into a binary tree structure. The first iteration yields the root node. Though it is not a problem to keep track of parent information throughout the binary tree, we only display the leaves in the simulator (as has been shown in Fig. 6). The first iteration will produce a root box with six faces. Each face is used for two grasp hypotheses parallel to the edges spanning it. See Fig. 9 for such a grasping pair. There are two types of approach techniques that we will apply for our experiments: a backup-grasp and a pinch-grasp. For the backup-grasp, each initial position is set to a constant distance from the face's center aligned to its normal. We let the hand approach the object along the normal until an arbitrary contact is detected. Afterwards, the hand retreats a small distance (the backup) to call the autograsping function. The autograsp is a built-in basic grasp of GraspIt! to uniformly close fingers and is therefore available for all hand models. Using the pinch grasp, we approximate the distance to the box center of the current face in order to force a grasp on the center of mass. This technique is assumed better for small part grasping, as the backup-grasp will usually retreat due to contact with another object first (e.g. a table under a pen). From the approximated distance, the autograsp is called. Note that the autograsp is the only final grasping technique that we use and that just one initial posture for each hand is applied. Furtheron, the boxes theirselves are not only transparent in the simulator, but also physically invisible for each object in the scene. Thus, the hand will grasp through the box representation and perform each grasp with contact on the real model. When all fingers are in contact, GraspIt! provides computation of two different grasp quality measures EPS_{L1} and VOL_{L1}.

After the grasps on the root box have been performed, we continue the proposed fit-and-split algorithm until it is finished. We collect all faces of boxes from the final approximation and remove some types of occluded, ungraspable faces from the set. Finally, the same grasping process is made for each remaining face as described above.

B. Experiments

For each of the models from the complete model data set (see Fig. 6), we go through this grasping evaluation for the root box and the boxes produced with gain parameters 0.90, 0.94 and 0.98. The boxes are computed according to the MVBB fit-and-split algorithm proposed in Section IV-C and then grasped with backup-grasp and pinch-grasp, respectively, which have been described in the last subsection. For each try, we take a look at the final grasp qualities and the on grasp that is best rated according to these quality measures. The results can be seen in Tab. III.

For each model, the overall faces describe the whole set of faces available from the current box set. It is clear that the number of overall faces is 12 times the number of boxes. Geometrical detection of blocked faces reduces the number of graspable faces drastically, as also the consideration of maximum width that the hand can grasp between its fingers. For example, there are no valid grasps for the Homer root box, as this is a large model (see best grasps in Fig. 10. The best way to grasp Homer is a pinch grasp on the highest splitting level (0.98), the same holds for the Bunny. In contrast, the Mug and Duck models are quite compact and best grasp qualities are found on a low splitting level (0.90). Additionally, the Duck proves very hard to handle, as it is a very small model. Half of the experiments for the Duck did not produce any force closure grasp.

Box set	Grasp	#valid faces	#overall faces	EPS_{L1}	VOL_{L1}
Duck Root	Backup	12	12	———	———
Duck Root	Pinch	12	12	0.00496	0.00879
Duck 0.90	Backup	14	36	———	———
Duck 0.90	Pinch	14	36	**0.03166**	**0.00274**
Duck 0.94	Backup	17	60	———	———
Duck 0.94	Pinch	17	60	———	———
Duck 0.98	Backup	33	108	0.00185	**0.00977**
Duck 0.98	Pinch	33	108	0.00161	0.00312
Mug Root	Backup	12	12	0.01213	0.00121
Mug Root	Pinch	12	12	0.01261	0.00144
Mug 0.90	Backup	12	24	0.00909	0.00043
Mug 0.90	Pinch	12	24	**0.01288**	**0.00288**
Mug 0.94	Backup	14	36	0.00682	0.00044
Mug 0.94	Pinch	14	36	———	———
Mug 0.98	Backup	19	48	0.00682	0.00033
Mug 0.98	Pinch	19	48	0.00420	0.00047
Bunny Root	Backup	12	12	0.01590	0.00670
Bunny Root	Pinch	12	12	———	———-
Bunny 0.90	Backup	20	24	0.01601	0.00420
Bunny 0.90	Pinch	20	24	0.00786	0.00200
Bunny 0.94	Backup	20	36	0.00767	0.00420
Bunny 0.94	Pinch	20	36	0.01356	0.00200
Bunny 0.98	Backup	51	120	**0.05652**	**0.00749**
Bunny 0.98	Pinch	46	120	0.01438	0.00336
Homer Root	Backup	12	12	———	———
Homer Root	Pinch	12	12	———	———
Homer 0.90	Backup	19	48	0.00721	0.00016
Homer 0.90	Pinch	15	48	0.00774	0.00011
Homer 0.94	Backup	20	60	0.00721	0.00016
Homer 0.94	Pinch	16	60	0.00774	0.00013
Homer 0.98	Backup	34	108	**0.00931**	**0.00046**
Homer 0.98	Pinch	29	108	0.00931	0.00012

TABLE III

TABLE OF THE EXPERIMENTAL GRASPING RESULTS.

VII. Discussion and Conclusion

In our approach, we combine several motivations known from the shape approximation and grasping literature. We prune the search space of possible approximations by rating and decomposing basic shapes. While Goldfeder *et al*. [10] use superquadrics as these basic shapes, their work confirms the expectation that planning on finer components is likely to find better grasps than returning the first stable grasp. This intuitively corresponds to the “grasping-by-parts” strategy. This strategy also underlies the presented approach of MVBB decomposition. However, Goldfeder *et al*. use superquadrics and the split-and-merge decomposition by Chevalier *et al*. [9], while we propose MVBBs and an efficient and valuable box decomposition. The fit-and-split strategy is motivated by work from Zha *et al*. [14] on superquadric shape splitting.

The trade-off of our approach lies in higher efficiency and simplicity for the price of precise shape approximation. However, we claim that exact approximation may not be necessary for grasping tasks. An evaluation of this claim is one of our next steps. Here, we take advantage of the basic box representation by using a very efficient splitting criterion (along 3 faces × 2 directions per MVBB). Additionally, boxes allow efficient further analysis, as there exist fast computational techniques on this representation (e.g. in terms of collision detection, neighborhood, etc.).

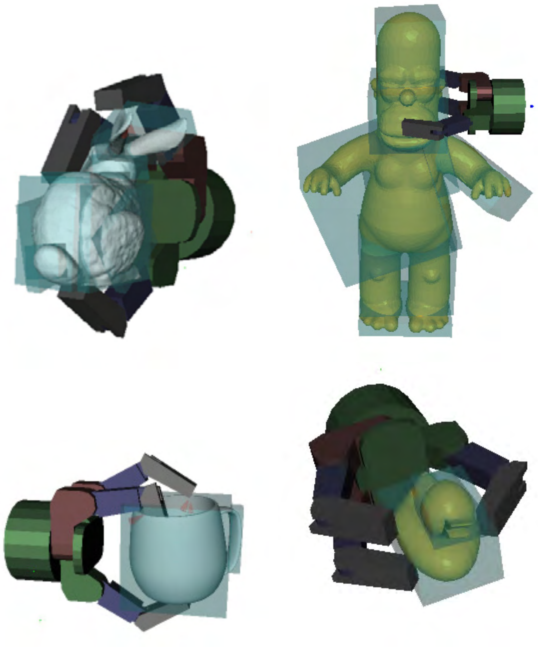


Fig. 10. Best grasps for the experimental grasping results in Tab. III.

This analysis will become important for another next step towards grasping objects based on the box representation. Managing valid grasps will not only be dependent on the box faces (which will be to prefer for a grasp), but also on the whole constellation of boxes. For example, not each face will be graspable in an arbitrary set of boxes due to occlusion.

Another issue is task dependency. There are different task types on which a grasp might depend. Just to pick up a cup and place it somewhere else might yield a different grasping action as picking up the cup to show it or hand it over. These grasp semantics might be mapped to boxes in the set, e.g.“grasp the largest box for a good force grasp to securely move the object”, “grasp the smallest box for a good pinch grasp to show a most unoccluded object to a viewer/camera” or “grasp a very outlying box so that another human/robot hand can overtake the object easily”. The latter semantics are quite valuable for applications that are based upon interacting with objects *before* the exploration and recognition stage (such as [18]).

As the presented approach is hierarchical, it is also possible to use dependencies between boxes and granularities of different hierarchical levels for shape approximation. Thus, the processing of shape approximation can be controlled and run parallel to the execution of a grasp.

Acknowledgements

This work was supported by EU through the project PACO-PLUS, IST-FP6-IP-027657. The authors would like to thank Michael Wünstel (Universität Bremen) for providing the raw partial scan data (Fig. 7).

References

[1] 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.

[2] Y. H. Liu, M. Lam, and D. Ding, “A Complete and Efficient Algorithm for Searching 3-D Form-Closure Grasps in Discrete Domain,” *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 805–816, 2004.

[3] K. Shimoga, “Robot Grasp Synthesis Algorithms: A Survey,” *International Journal of Robotic Research*, vol. 15, no. 3, pp. 230–266, 1996.

[4] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, “Automatic Grasp Planning Using Shape Primitives,” in *IEEE International Conference on Robotics and Automation (ICRA ’03)*, 2003, pp. 1824–1829.

[5] A. Morales, E. Chinellato, A. H. Fagg, and A. P. del Pobil, “Using experience for assessing grasp reliability,” *International Journal of Humanoid Robotics*, vol. 1, no. 4, pp. 671–691, 2004.

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

[7] F. Solina and R. Bajcsy, “Recovery of Parametric Models from Range Images: The Case for Superqaudrics with Global Deformations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 2, pp. 131–147, 1990.

[8] G. Biegelbauer and M. Vincze, “Efficient 3D Object Detection by Fitting Superquadrics to Range Image Data for Robot’s Object Manipulation,” *IEEE International Conference on Robotics and Automation*, 2007.

[9] L. Chevalier, F. Jaillet, and A. Baskurt, “Segmentation and Superquadric Modeling of 3D Objects,” *Journal of Winter School of Computer Graphics, WSCG’03*, 2003.

[10] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, “Grasp Planning Via Decomposition Trees,” in *IEEE International Conference on Robotics and Automation*, 2007.

[11] D. Katsoulas, “Reliable Recovery of Piled Box-like Objects via Parabolically Deformable Superquadrics,” in *Proc. of the 9th IEEE International Conference on Computer Vision*, vol. 2, 2003, pp. 931–938.

[12] D. Aarno, J. Sommerfeld, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft, and N. Krüger, “Early Reactive Grasping with Second Order 3D Feature Relations,” in *Proc. of the Int. Conference on Robotics and Automation, Workshop: From features to actions - Unifying perspectives in computational and robot vision*, 2007, pp. 319–325.

[13] J. Tegin, S. Ekvall, D. Kragic, B. Iliev, and J. Wikander, “Experience based Learning and Control of Robotic Grasping,” in *Workshop: Towards Cognitive Humanoid Robots at the IEEE-RAS International Conference on Humanoid Robots*, 2006.

[14] H. Zha, T. Hoshide, and T. Hasegawa, “A Recursive Fitting-and-Splitting Algorithm for 3-D Object Modeling Using Superquadrics,” in *Proceedings of the Fourteenth International Conference on Pattern Recognition*, vol. 1, 1998, pp. 658–662.

[15] 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.

[16] S. Gottschalk, M. C. Lin, and D. Manocha, “OBBTree: A Hierarchical Structure for Rapid Interference Detection,” *Computer Graphics*, vol. 30, no. Annual Conference Series, pp. 171–180, 1996.

[17] W. T. Townsend, “The BarrettHand Grasper – Programmably Flexible Part Handling and Assembly,” *Industrial Robot: An International Journal*, vol. 27, no. 3, pp. 181–188, 2000.

[18] A. Ude, K. Welke, J. Hale, and G. Cheng, “Data Acquisition for Building Object Representations: Discerning the Manipulated Objects from the Background,” in *Unpublished*, 2007.

Exploiting Similarities for Robot Perception

Kai Welke* , Erhan Oztop†‡ , Gordon Cheng†‡ and Rüdiger Dillmann*
*University of Karlsruhe (TH), IAIM, Institute of Computer Science and Engineering (CSE)
P.O. Box 6980, 76128 Karlsruhe, Germany
†JST, ICORP, Computational Brain Project
4-1-8 Honcho, Kawaguchi, Saitama, Japan
‡ATR Computational Neuroscience Lab., Dept. of Humanoid Robotics and Computational Neuroscience
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan

Abstract— **A cognitive robot system has to acquire and efficiently store vast knowledge about the world it operates in. To cope with every day tasks, a robot needs to learn, classify and recognize a manifold of different objects. Our work focuses on an object representation scheme that allows storing perceived objects in a compact way. This will enable the system to store extensive information about the world and will ease complex recognition tasks. The human visual system deploys several mechanisms to reduce the amount of information. Our goal is to develop an artificial system that mimics these mechanisms to create representations that can be used in cognitive tasks. In particular, in this paper we will present an approach that exploits similarities among different views of objects. The proposed representation scheme allows for reduction of storage required for the representation of objects and preserves the information about the similarity among objects. This is achieved by selecting 'important views' of objects, depending on their stability. Furthermore, by extending the same approach to multiple objects, we are able to exploit similarities between objects to find a common representation and to further reduce the storage requirements.**

I. Introduction

The main focus of our work is to develop an object representation and learning scheme, that is suitable for learning in humanoid robots, e.g. via action-perception coupling, much the same way as humans learn about objects in their environment. To achieve the ability of recognizing objects from all viewing directions, we introduced a learning and representation scheme that allows generalizing to specific views of objects [1]. We have shown that, given locations of important views, the objects can be represented in a depth rotational invariant manner, with a reduced amount of views stored as representations. However, in the former work the important views were selected manually.

In this paper, we introduce a solution on how to select specific important views of objects automatically. Our approach is driven by the observation that there are specific views of an object, that allow recognizing a wide range of rotational variations of the object. Such views are often referred to as *stable* views [2]. With these stable views of an object, its appearance can be described using a minimal set of views.

The robot perceives the world in different modalities, depending on the sensors and feature extraction methods used. In real world scenarios, the robot will face objects, which are similar in at least one modality and are only separable by combining different modalities. Furthermore, learning of objects from all possible viewing directions will reveal even more similarities between views of different objects. In our approach, we identify views that are shared between objects. Similar views can be subsumed and stored only once. In such cases we do not want to recognize objects in the modality, in which the similarities exist, but rather aim at a representation that preserves the information, which objects are candidates for the specific view and modality. The ability to discriminate such objects has to be achieved by combining multiple modalities. The shared view of objects in one modality can then be used to restrict the possibilities in other modalities to only a few objects.

Our approach follows a global appearance-based representation scheme of objects. In appearance-based vision systems, objects are represented with multiple retinal projections of object views. In contrast, model-based representations need more structural information, like full 3D models, which are hard to aquire during online learning [3]. Furthermore, we use global object descriptors to identify important views of the object. The majority of recent work on object recognition uses local features, which describe important locations in the object's appearance, considering measures for texturedness or cornerness. These systems perform well in real environments, are able to handle occlusion, and usually offer invariances to at least shift and rotation in the camera plane [4] [5]. Recognition of all rotations of an object with local features is possible, but impractical in terms of efficiency due to the amount of stored local feature representations.

The selection of important views of an object has strong correspondence to the notion of canonical views used in psychology [6]. In the past, different criterias that define canonical views have been introduced. Blanz et al. give a good overview of different criteria [7]. As our aim is to implement an object recognition system based on our representation and learning scheme, we are mainly interested in the *goodness for recognition* criteria. More precisely, we identify views of the objects, that are stable for at least small transformations.

While the work on canonical views copes largely with one outstanding view of the object, a representation scheme which is used for recognition has to rely on multiple important views of the object, which together should cover the

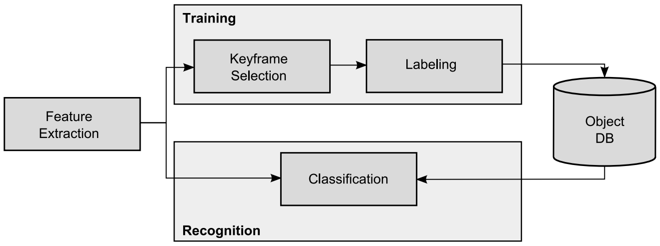


Fig. 1. System structure of the proposed learning and recognition system.

complete appearance.

Hall et al. presented an approach to extract multiple important views of an object by identifying the most unique views of the object [8]. The identification of unique views is suitable, if the objects are to be visualized or if the resulting views are used only to discriminate objects. The approach did not take into account the similarity of views. Moreover, the resulting views did not capture similarities among different objects.

Yamauchi et al. introduced an approach for the identification of important views which combines the saliency of views and the stability criterion [2]. They proposed an approach based on spherical graphs which reflect all available viewing directions of the object. Important views were identified using Zernike Moments [9] to measure the similarity between neighbored views in the spherical graph. In their work, Yamauchi et al. did not take into account similarities among different objects. Each object had its own set of keyframes regardless of the appearance of other objects. Furthermore the number of extracted views per object had to be predefined.

In the following, we present an approach that can be applied to the output of different feature extraction methods. Our method will identify stable views for the objects considering the Euclidian distance of the output from the used extraction method. In the following we will referr to these stable views as *keyframes* . Furthermore, our method exploits similarities among different objects. The number of keyframes per object does not have to be predefined. Rather, the accuracy can be defined with an overall maximum error, thus the system generates a different number of keyframes per object, depending on the object's appearance.

II. System description

A. Overview

Figure 1 gives a schematic overview of the system structure used throughout this paper. The system can be divided into a training part and a recognition part. In the application on a robot system, both parts have to be executed simultanously to allow the acquisition of new objects during interaction with the environment. The following sections will primarily focus on the training part, since the recognition part needs to rely on more than one modality as explained later (subsection II-E).

As mentioned earlier, the presented approach does not depend on a certain feature extraction method. The extraction method used should fulfill the following requirements:

- The extraction method has to capture the global appearance of an object.
- The extraction method should represent each view invariant to rotations in the viewing plane.
- The extracted feature vectors should be of reasonable size to allow fast extraction of keyframes.

For the experiments in this paper, we use color cooccurrence histograms (CCHs) to extract descriptors of the global appearance of views. The extraction of CCHs will be explained in subsection II-B.

During keyframe selection, significant views of the objects are identified by clustering the feature space into classes containing similar views. Each class is identified with its centroid, which is referred to as a keyframe.

Objects are assigned to keyframes in the labeling step. Each keyframe will be associated with all objects that have views in the corresponding class. Furthermore, we define the activation of a keyframe as the number of views an object participates with in the corresponding class. The keyframes are stored in the object database, together with the labels and activations determined in the labeling phase.

During recognition, the extracted features are classified using the stored keyframes from the object database. The classification will output all objects that have views similar to the current percept and the corrsponding activations.

The following subsections will explain the different elements of the system structure in detail.

B. Feature Extraction

Throughout this paper, we will use color cooccurrence histograms (CCHs) for the description of object appearances. CCHs were chosen because they offer some properties which allow the application in real world recognition tasks. For instance CCHs offer a description of the object, which is invariant to the rotation in the viewing plane, when the parameters are chosen accordingly. Furthermore, CCHs offer some robustness towards scaling. Finally, CCHs combine texture information (in terms of information about pairs of neighbored pixels) as well as color information.

Based on work performed by Haralick et al. [10], CCHs were first introduced by Chang et al. [11]. In their work they define an entry in the color coocurrence histogram by the cooccurring colors and their distances in an observed image:

$$CH(c_1, c_2, \Delta x, \Delta y), \tag{1}$$

where c_1 and c_2 describe two colors in RGB space and Δx and Δy describe their distances in terms of pixels in the observed image. To achieve a rotation invariant description, only the absolute distance of the two colors is used in their approach:

$$d = \sqrt{(\Delta x)^2 + (\Delta y)^2} \tag{2}$$

The cooccurrence histogram is derived by counting all occurrences of entries CH in the observed images.

In our implementation only cooccurences with a distance $d < 1.5$ are observed. This restricts the cooccurences to

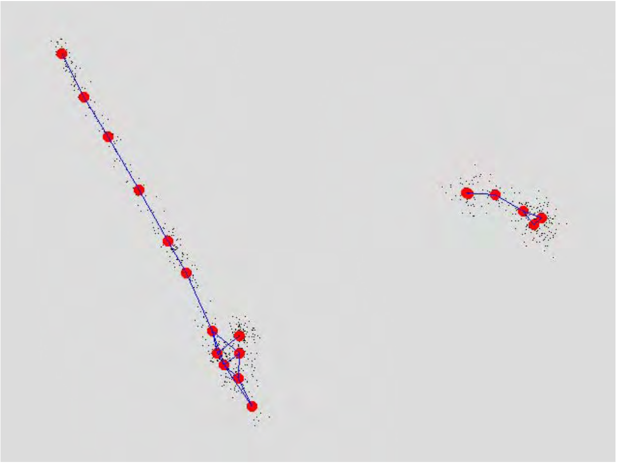


Fig. 2. Resulting growing neural gas network after 2000 iterations. 10 objects with 72 views each were used as input. The features and node positions were projected into 2D eigenspace.

neighboring pixels. Furthermore, the red and green color channels (I_r, I_g) and the gradient magnitude of both color channels $(\nabla I_r, \nabla I_g)$ are used as histogram dimensions. The choice of these image descriptors is motivated by the previous works of Ekvall et al. [12]. They showed that with the combination of intensity and gradient descriptor calculated on the basis of the red and green channels good recognition results could be achieved. The cooccurrences in each channel are considered separately. Each channel is quantized to 80 clusters in a preprocessing step. This results in a feature vector of 320 dimensions, which is still of reasonable size.

C. *Keyframe Selection*

The identification of similar views in the set of CCH features can be achieved by clustering the feature space into similar classes. Since the keyframe selection process will run autonomously, an unsupervised clustering method is required for our approach. Furthermore, the applied method should select the number of generated clusters dependent on the distribution of the input data rather than on a prespecified number of keyframes. One algorithm that fulfills these requirements is the Growing Neural Gas algorithm (GNG) which was first introduced by Fritzke [13]. The GNG is a self organizing map, which grows in the process of training according to the distribution of the input data. Thus the GNG algorithm creates a topological map which represents the distribution of the training data.

The GNG algorithm combines the Competitive Hebbian Learning and the Neural Gas method proposed by Martinez et al. [14] with an incremental learning approach. GNG thus overcomes the problem of prespecifying the number of nodes that is required to reach a certain goal. Heinke et al. [15] provided a comparison of different incremental neural network algorithms. Their comparison comprises Growing Cell Structures (GCS), Fuzzy Artmap (FAM), and Growing Neural Gas (GNG). As benchmark the performance of the multi-layered perceptron (MLP) was used. The GNG algorithm outperforms FAM on nearly all datasets and generates less nodes then GCS with similar performance for most datasets.

In the following, a brief introduction to the GNG algorithm is given to ease the understanding of the choice of certain parameters and the termination criterion. For a more detailed description of the algorithm the reader is referred to [13]. The network consists of the following components:

- A set of nodes N, each node $n \in N$ has an associated position vector w_n.
- A set of edges E, each edge $c \in E$ connects pairs of nodes and has an associated age.

The algorithm can be described with the following steps:

1) Create two nodes n_1 and n_2 with random positions w_{n_1} and w_{n_2}.
2) Select one feature f from the training set randomly.
3) Identify the nearest and second nearest nodes n_a and n_b to the feature f.
4) Increment the age of all edges starting from n_a.
5) Accumulate the error of node n_a by the squared distance of node position w_{n_a} and input signal f:
$$\Delta e_{n_a} = ||w_{n_a} - f||^2$$
6) Move the nearest node n_a and the second nearest node n_b towards the input signal f using the learning rates sp_a and sp_b:
$$\Delta w_{n_a} = sp_a(f - w_{n_a})$$
$$\Delta w_{n_b} = sp_b(f - w_{n_b})$$
7) Reset the age of the edge from the nearest to second nearest node c_{n_a,n_b} to zero. If no edge exists, create a new edge.
8) Remove edges with an age larger than a_{max}.
9) If the accumulated error e_{n_e} of one node n_e exceeds the maximum error e_{max} insert a new node in the following way:
   - Identify the node connected to n_e with the maximum error n_m.
   - Insert a new node n_c halfway between the two nodes n_e and n_m:
   $$w_{n_c} = \frac{(w_{n_e} + w_{n_m})}{2}$$
   - Insert edges c_{n_c,n_e} and c_{n_c,n_m} and remove the edge c_{n_e,n_m}.
   - Set the accumulated error e_{n_c} of the new node to the mean error of the nodes n_e and n_m.
   - Decrease the accumulated error e_{n_e} and e_{n_m} by multiplication with a constant factor $\alpha < 1$.
10) Decrease all error variables by multiplication with a constant $\gamma < 1$:
$$e'_n = \gamma e_n$$
11) Check termination criterion. If not matched restart with step 2.

Depending on the termination criterion and the parameters used for training, the GNG algorithm will produce a topological map of the input data with respect to the distribution of the input data. Figure 2 shows an example outcome of the GNG clustering for 720 CCH features, which describe the rotation of 10 objects.

The parameters used for training the GNG were determined empirically. Aim of the parameter choice was a balance between stability of the network and fast convergence. Throughout the experiments a maximum edge age $a_{max} = 20$ was used. The learning rates were set to $sp_a = 0.16$ and $sp_b = 0.01$. The factors for the adjustment of the accumulated error in the case of a new node (α) and for each iteration (γ) were set to $\alpha = 0.001$ and $\gamma = 0.995$.

The parameters for the maximum accumulated error per node e_{max} and the termination criterion directly influence the number of nodes created for the input data. The choice of these parameters will be discussed in section III.

Each node from the network represents one cluster in the space of input features and is considered a keyframe.

D. Labeling

The clustering results in a set of nodes $N = (n_1, \ldots, n_r)$. In order to use these nodes for object representation and recognition we have to restore the association of object views with the clusters formed by the nodes. In the following, s objects $W = (F_1, \ldots, F_s)$ each described with t features $F_x = (f_{x,1}, \ldots, f_{x,t})$ are considered.

To associate object labels with nodes all objects x and their features $f_{x,v}$ are traversed. For each node n_i the number of features of the object where the node is the nearest neighbor to the corresponding feature is determined as:

$$b_{i,x} = |\{f_{x,v} : i = argmin_{u \in \{1,\ldots,r\}} ||w_{n_u} - f_{x,v}||^2\}| \quad (3)$$

If $b_{i,x}$ is not zero, the object label x is appended to the list of object labels L_i for node n_i, if not already present:

$$L'_i = (L_i, x) \quad (4)$$

Additionally, the activation $a_{i,x}$ of the node n_i for the object x is calculated by the following equation:

$$a_{i,x} = \frac{b_{i,x}}{\sum_x b_{i,x}} \quad (5)$$

The activation describes how likely a feature which is associated to the node n_i will belong to the object x. If $b_{i,x}$ is non-zero, the activation $a_{i,x}$ is appended to the list of activation A_i of the node:

$$A'_i = (A_i, a_{i,x}) \quad (6)$$

It is guaranteed that for all labels of one object the sum of the corresponding activations is equal one, i.e.:

$$\sum_{x=1}^{s} a_{i,x} = 1 \quad (7)$$

This shows that if the activation for an object for the node equals 1, then the corresponding keyframe describes one object uniquely. The node will only contain one object label in this case.

In the object database, the node positions $w_{n_1}, \cdots, w_{n_r}$ are stored along with the associated labels $L_{n_1}, \cdots, L_{n_r}$ and activations $A_{n_1}, \cdots, A_{n_r}$.

E. Classification

In the classification step, a perceived view of an object in terms of its CCH f is matched with the keyframes stored in the object database. This can be accomplished by identification of the nearest neighbor n_i in the set of keyframes:

$$i = argmin_{u \in \{1,\ldots,r\}} ||w_{n_u} - f||^2 \quad (8)$$

If the label list L_i contains only one label, the corresponding object is found. Otherwise the classification can not be performed in a unique way. The list of labels L_i contains objects that have views similar to the currently perceived view. The corresponding activations A_i describe the probabilities for the individual objects.

In the case of multiple potential candidates for the current view, the feature extraction method used is not sufficient to separate between the objects in this class. In this case other modalities are required to uniquely detect the object corresponding to the perceived view. For this purpose our approach reduces the number of possibilities to similar objects in the modality observed and allows the restriction to only a few objects for the search in other modalities.

III. Parameter Evaluation

For all experiments in this paper, object views from the Amsterdam Library of Object Images (ALOI) [16] are used. The ALOI contains images of objects on black background from 72 distinct viewing angles, which are generated by rotating the object around the vertical axis. We use 10 objects for the evaluation of our approach, which results in 720 CCHs.

As mentioned earlier, the maximum error e_{max} and the termination criterion are crucial for the number of nodes that are generated by the GNG algorithm. In the following, our choice of these parameters is explained.

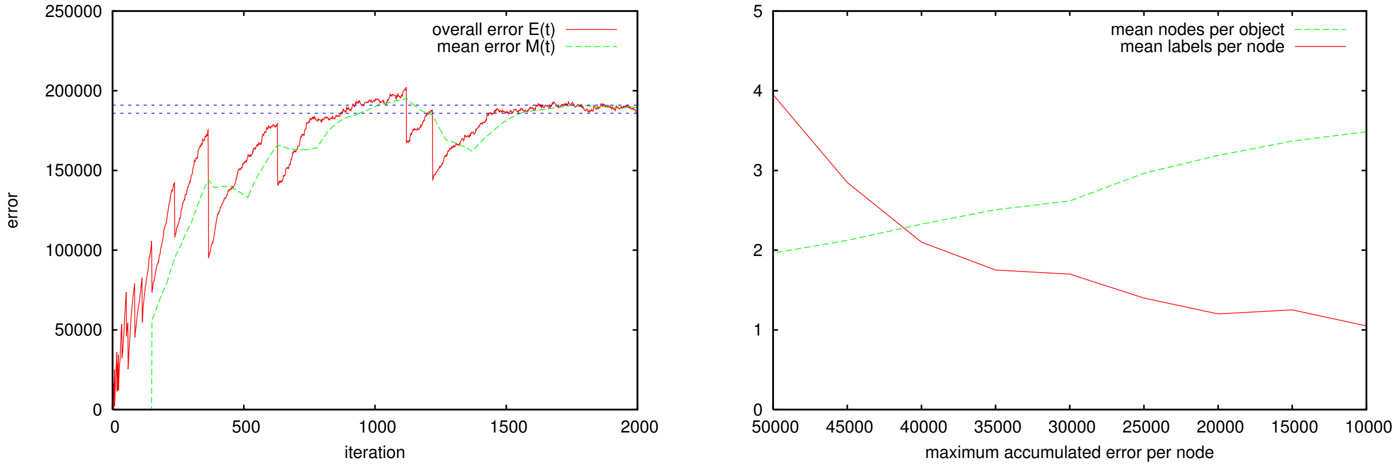
To verify if the network has converged, the overall error $E(t)$ of the network is monitored for each iteration t. The overall error can be determined by summing up the accumulated errors of all nodes:

$$E(t) = \sum_{x=1}^{r} e_{n_x}(t) \quad (9)$$

The overall error is smoothed by calculating the mean overall error $M(t)$ over the last 200 iterations. This helps in coping with local peaks in the course of the error over the iterations. To detect the convergence of the network, we check if $M(t)$ is in a defined range r for a minimum number of iterations Δt. The termination criterion $c(t)$ is defined in the following way:

$$c(t) = \begin{cases} 0 & a \leq M(t - t_0) < b; 0 \leq t_0 < \Delta t; b - a < r \\ 1 & \text{otherwise} \end{cases}$$

We choose a range of $r = 5000$ and set the minimum time the mean overall error has to stay in this range to $\Delta t = 500$ iterations. Figure 3(a) shows the development of the overall network error $E(t)$ and the mean error $M(t)$ during one training phase. Every time the accumulated error



(a) Overall network error during one training phase. The upper and lower bounding from the termination criterion are denoted with horizontal lines.

(b) Development of the mean number of labels per node and the mean number of nodes per object dependent on the maximum accummulated error e_{max}.

Fig. 3. Results from the parameter evaluation

of one node exceeds e_{max} the accumulated error is adjusted and a new node is inserted. This results in a diminution of the overall error. On new input data, the accumulated error of both nodes increases again. The overall error of a network containing more nodes can exceed the overall error of a network with less nodes because each single node can accumulate an error of up to e_{max}. The termination criterion terminates the training, if the error stays inside the range r denoted by the two horizontal lines.

In order to determine the maximum node error e_{max} for our experiments, two measures were observed using a range of $e_{max} \in [10000 : 50000]$. First the mean number of labels per node $\overline{L}$ was observed. Furthermore, the mean number of labels per object $\overline{I}$ was observed. In figure 3(b) both measures $\overline{L}$ and $\overline{I}$ are recorded. The graphs show that with a large e_{max}, the mean number of labels per node decreases fast. With decreasing e_{max}, the gradient of $\overline{L}$ reduces. The number of nodes per object $\overline{I}$ grows about linear with decreasing e_{max}. A suitable choice of e_{max} should reduce the number of labels produced per node, since this decreases the uncertainty during recognition. Furthermore, not too many nodes per object should be generated, since the resulting representation has to be compact. For this reason, a maximum accumulated error of $e_{max} = 25000$ was chosen for our experiments. The choice of a maximum accumulated error less than 25000 would result in the generation of more nodes without significantly decreasing the number of labels per node.

IV. Experimental Results

In our experiments, the GNG algorithm proved to be very stable. For the 10 objects with 720 views the number of nodes usually converged to 19. Depending on the sequence of the random selection of input data that was exposed to the network, occasionally 18 or 20 nodes were created.

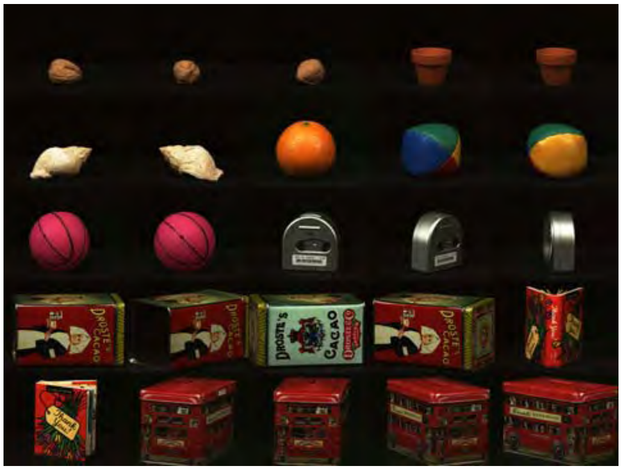


Fig. 4. Important views of objects as extracted by our approach. Only views corresponding to an activation $a_{i,x}$ above 20% are shown.

A. Keyframe Selection

The GNG network produces clusters of similar views and corresponding nodes with object labels L_i and activations A_i. The node positions do not exactly correspond to object views. In order to visualize important views for the objects, the nearest neighbor from the set of samples for each object x which is in the list of labels L_i is identified. Views are reported only if the activation $a_{i,x}$ from the list of activations A_i is above a given threshold. Thus, only those views are reported that are produced by clusters where the object participates with a significant amount of views.

Figure 4 shows the important views produced by our approach with a threshold of $a_{i,x} > 0.2$. The selected views depend on the used feature extraction method. Using a color descriptor like CCH results in the selection of views which are stable in terms of color.

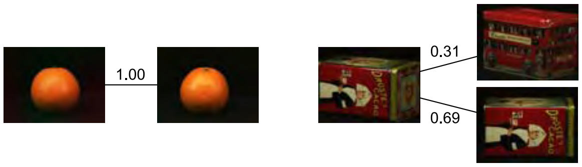


Fig. 5. During recognition, the orange on the left side can be identified uniquely. The can on the right side is associated to a keyframe with two labels. The connections are tagged with the corresponding activations $a_{i,x}$.

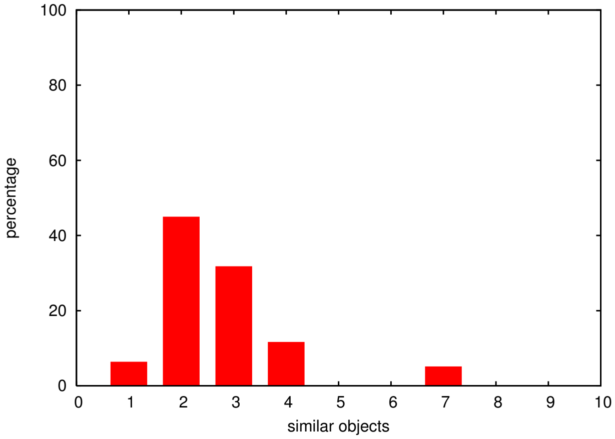


Fig. 6. Percentage of views of all 10 objects in relation to the number of similar objects associated.

B. Recognition

In the recognition phase all object views are associated to the corresponding keyframes. Figure 5 shows two examples of associated views. In the first case, the view was associated to a keyframe which containes only one label. In the second case, the keyframe contained two labels. The keyframes are visualized with the corresponding closest views for each label contained in the label list. The connections are tagged with the activations $a_{i,x}$.

In order to provide a measure on how our approach reduces the uncertainty about the perceived object, we associate all object views to their keyframes. For each view the uncertainty can be expressed with the number of similar objects obtained from the label list. Figure 6 shows the percentage of views in relation to the number of similar objects. 6% of the object views are associated to keyframes which contain only one view and thus can be uniquely identified. 80% of the views are associated to keyframes which contain two or three labels. The remaining views are associated to keyframes with four and more views. The mean number of similar objects per view is about 2.7.

V. Conclusion

The proposed approach allows for the extraction of keyframes on the basis of similarities among objects. For 10 objects with overall 720 views we were able to reduce the number of stored features for one modality to only 19. The experiments show, that with these 19 features, the potential candidates for a perceived object can be reduced to 2.7 on average.

An artificial perception system for a cognitive robot has to rely on more then one modality to identify and classify the manifold of different object types it encounters in real world tasks. The proposed approach will be used in conjunction with a combination of different descriptors for the object appearances. Despite the mentioned CCHs we plan to apply the same approach to other feature extraction methods eg. Zernike Moments. If chosen accordingly, the combination of different modalities will allow to identify the perceived objects uniquely.

Finally, the system will be implemented on our humanoid robot to ease the acquisition of objects during exploration of the environment.

Acknowledgment

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] K. Welke, E. Oztop, A. Ude, R. Dillmann, and G. Cheng, “Learning features for an object recognition system,” in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Genova, Italy, 2006, pp. 290–295.

[2] H. Yamauchi, W. Saleem, S. Yoshizawa, Z. Karni, and H.-P. Seidel, “Towards stable and salient multi-view representation of 3d shapes,” in *Proceedings of the International Conference on Shape Modeling and Applications*, 2006.

[3] V. Lepetit, L. Vacchetti, D. Thalmann, and P. Fua, “Fully automated and stable registration for augmented reality applications,” in *Proceedings of the Second IEEE and ACM Symposium on Mixed and Augmented Reality*, 2003, pp. 93–102.

[4] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1999, pp. 1150–1157.

[5] K. Welke, P. Azad, and R. Dillmann, “Fast and robust feature-based recognition of multiple objects,” in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, Genova, Italy, 2006, pp. 264–269.

[6] S. Palmer, E. Rosch, and P. Chase, “Canonical perspective and the perception of objects,” in *Attention and Performance IX*, J. Long and A. Baddeley, Eds. Hillsdale, NJ: Erlbaum, 1981.

[7] V. Blanz, M. J. Tarr, H. H. Bülthoff, and T. Vetter, “What object attributes determine canonical views?” Max Planck Institute for Biological Cybernetics, Tech. Rep. Technical Report No. 42, 1996.

[8] P. Hall and M. Owen, “Simple canonical views,” in *Proceedings of the british machine vision conference*, 2005, pp. 7–16.

[9] S.-H. Kim, I.-C. Kim, and I.-S. Kweon, “Probabilistic model-based object recognition using local zernike moments,” in *IAPR workshop on Machine Vision Applications*, Nara , Japan, Dec. 2002.

[10] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural features for image classification,” in *IEEE Transactions on Systems, Man and Cybernetics*, 1973, pp. 610–621.

[11] P. Chang and J. Krumm, “Object recognition with color cooccurrence histogram,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1999.

[12] S. Ekvall and D. Kragic, “Receptive field cooccurrence histograms for object detection,” in *Proceedings of the IEEE/RSJ International Conference Intelligent Robots and Systems*, 2005.

[13] B. Fritzke, “A growing neural gas network learns topologies,” in *Advances in Neural Information Processing Systems*, vol. 7, 1995.

[14] T. Martinez and K. Schulten, “Topology representing networks,” in *Neural Networks*, vol. 7, 1994, pp. 507–522.

[15] D. Heinke and F. H. Hamker, “Comparing neural networks: A benchmark on growing neural gas, growing cell structures, and fuzzy artmap,” in *IEEE Transactions on Neural Networks*, vol. 9, 1998.

[16] J. M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, “The Amsterdam library of object images,” *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103–112, 2005.

International Journal of Humanoid Robotics
© World Scientific Publishing Company

Birth of the Object: Detection of Objectness and Extraction of Object Shape through Object Action Complexes

Dirk Kraft, Emre Başeski, Mila Popovic, Norbert Krüger
University of Southern Denmark, Odense, Denmark
{kraft,emre}@mmmi.sdu.dk, mipop05@student.sdu.dk, norbert@mmmi.sdu.dk

Nicolas Pugeault
University of Edinburgh, Edinburgh, UK
npugeaul@inf.ed.ac.uk

Danica Kragic
Royal Institute of Technology, Stockholm, Sweden
dani@kth.se

Sinan Kalkan, Florentin Wörgötter
BCCN, University of Göttingen, Göttingen, Germany
{sinan,worgott}@bccn-goettingen.de

We describe a process in which the segmentation of objects as well as the extraction of the object shape becomes realized through active exploration of a robot vision system. In the exploration process, two behavioural modules that link robot actions to the visual and haptic perception of objects interact. First, by making use of an object independent grasping mechanism, physical control over potential objects can be gained. Having evaluated the initial grasping mechanism as being sucessful, a second behaviour extracts the object shape by making use of prediction based on the motion induced by the robot. This also leads to the concept of an 'object' as a set of features that change predictably over different frames.

The system is equipped with a certain degree of generic prior knowledge about the world in terms of a sophisticated visual feature extraction process in an early cognitive vision system, knowledge about its own embodiment as well as knowledge about geometric relationships such as rigid body motion. This prior knowledge allows for the extraction of representations that are semantically richer compared to other approaches.

Keywords: Early Cognitive Vision, Grasping, Exploration

1. Introduction

According to Gibson[1] an object is characterized by three properties: It

O1 has a certain minimal and maximal size related to the body of an agent
O2 shows temporal stability
O3 is manipulatable by the agent

Note that all these three properties are defined in relation to the agent (even temporal stability (O2) is relative to the lifetime span). Hence, no general agent independent criterion can be given. For an adult, a sofa certainly fulfills all three properties but for a fly, a sofa is more a surface than an object.

The detection of 'objectness' according to the three properties described above is not a trivial task. When observing a scene, usually in a visual system, a number of local features become extracted for which it is unclear whether and to which object they correspond to. Actually, property (O3) can only be tested by actively acting on the scene in case that no prior object knowledge is available.

In many artificial systems, in particular in the context of robotics, the object shape is given by a CAD representation a priori and is then used for object identification and pose estimation (see, e.g., Lowe[2]). However, CAD representations are not available in a general context, and for any cognitive system, it is an important prerequisite that it is able to learn object representations from experience.

In this paper, we address both problems: We introduce a procedure in which the objectness becomes detected based on the three Gibsonian criteria mentioned above. In addition, the object shape becomes extracted by making use of the coherence of motion induced by the agent after having achieved physical control over something that might turn out to become an object.

Our approach is making use of the concept of Object Action Complexes (OACs) where we assume that objects and actions (here the "grasping action" and controlled object movement) are inseparably intertwined. Hence, the intention of performing a grasp, the actual attempt to grasp and the evaluation of its success as well as a controlled movement of the object in case of a successful grasp will let the 'objectness' as well as a representation of the object's shape emerge as the consequence of the actions of the cognitive agent[a].

It is worth noting that both aspects, achieving physical control over a *thing*[b] as well as the extraction of object shape is based on a significant amount of prior knowledge, which however is much more generic than a CAD model of an object. More specifically, this prior consists of the system's knowledge about

1) its own body in terms of the shape, the degrees of freedom and the current joint configuration of the robot arm as well as the relative position of the stereo camera system and the robot co-ordinate system,
2) a developed early cognitive system[3] that extracts local multi-modal symbolic descriptors (see figure 1, a–e), in the following called primitives, and relations defined upon these primitives expressing statistical and deterministic properties of visual information (see figure 2).

[a] We note that this extends the notion of "affordances" by Gibson. According to Gibson: Objects afford actions. While this remains true, it is also - in our hands - the case that an action defines an object. Hence the action of drinking defines a cup, where the action of "placing on top" makes the same(!) thing a pedestal (an upside down cup).

[b] We denote with 'thing' something that causes the extraction of a visual feature but which is not yet characterized as an object since it could be for example also something fixed in the workspace of the robot and hence does not fulfill condition three above.

3) two behavior modules in terms of two OACs:

B1 An object independent 'grasping reflex' leads in some cases to successful grasping of potential objects (figure 1e shows the end-effector's pose for one successful grasp). Note that here it is less important to have a high success-rate of grasping attempts but that is is more important that a success is actually *measurable* and that it then triggers a second exploration mechanism (see B2).
The 'grasping reflex' is based on three semantic relations defined within the early cognitive vision system: First, co-planarity of descriptors indicate surfaces and by that possible grasping options. The co-planarity relation is enhanced by a co-linearity and co-colority relation to further enhance the success rate of the 'grasping reflex'.

B2 After a successful grasp an accumulation module explores the object by looking at different views of the object (see figure 1f,g) and accumulating this information to determine the objectness of the thing as well as to extract the shape of the object 1h. This accumulation module is based on prediction based on a rigid body motion relation between primitives. Having gained physical control over an object by the grasping reflex allows for inducing a rigid body motion on the object and by that the object (its objectness as well as its shape) can be characterized by the set of visual descriptors changing according to the induced motion.

The idea of taking advantage of active components for vision is in the spirit of active vision research[4,5]. The grounding of vision in cognitive agents has been addressed for example by a number of groups in the context of grasping[6,7] as well as robot navigation[8].

The work of Fitzpatrick and Metta[6] is the most related one to our approach since the overall goal as well as the hardware set up is similar: Finding out about the relations of actions and objects by exploration using a stereo system combined with a grasping device. We see the main distinguishing feature of this work to our approach in the amount of pre-structure we use. For example, we assume a much more sophisticated vision system that covers multiple visual modalities in a condensed form as well as visual relations defined upon them. This allows us to operate in a highly structured feature space where, instead of pixel–wise representations, we can operate on local symbols for which we can predict changes not only of position but also other feature attributes such as orientation and colour. Furthermore, the use of a very precise industrial robot allows for a precise generation of changes exploited for the extraction of the 3D shape of the object. It is not clear what exact prior knowledge can be assumed in the human system. However, there exist strong indications for an innate concept of 3D space as well as for sophisticated feature extraction mechanisms being in place very early in visual experience. For a discussion of this issue see for example Kellmann and Arterberry[9]). The question of prior knowledge in the context of depth perception and possible consequences for the design of artificial systems is described in Krüger and Wörgötter[10].

Similar to Fitzpatrick and Metta[6], we assume first 'reflex-like' actions that triggers exploration. However, since in our system the robot knows about its body and the 3D geometry of the world and since the arm can be controlled more precisely, these reflexes can make use of more complex visual events. As a consequence we can make use of having physical control over the object and to extract rather precise 3D information (in addition to the appearance based information coded in the primitives).

Modayil and Kuipers[8] addressed the problem of detection of objectness and the extraction of object shape in the context of a mobile robot using laser information. Here also motion information (in terms of the odometry of the mobile robot) is used to formulate predictions. In this way they were able to extract a top-view of the 3D shape of the object however only in terms of geometric information and only in terms of a 2D projection to the ground floor.

The paper is organized as following: In section 2 the early cognitive vision system is briefly described. In section 3 and 4 we give a description of the two sub-modules, i.e., the grasping reflex and the accumulation scheme. Sub–aspects of the work have been presented at two workshops[11,12].

2. An Early Cognitive Vision System

In this section, we introduce the visual system in which the detection of 'objectness' as well as the acquisition of the object representation is taking place. The system is characterized by rather structured prior knowledge: First, a scene representation is computed in terms of local symbolic descriptors (in the following called primitives) covering different visual modalities as well as 2D and 3D aspects of visual data (section 2.1). Second, there are relations defined upon the symbolic descriptors that cover spatial and temporal dependencies as briefly described in section 2.2. It is only the use of this prior knowledge that allows for the formulation of the two OACs described in sections 3 and 4.

2.1. *Multi–modal Primitives as local scene descriptors*

In this work we use local, multi–modal contour descriptors hereafter called *primitives*[3,13] (see figure 1). These primitives give a semantically meaningful description of a local image patch in terms of position as well as the visual modalities orientation, colour and phase. The importance of such a semantic grounding of features for a general purpose vision front–end, and the relevance of edge–like structures for this purposes was discussed, e.g., by Elder[14].

The primitives are extracted sparsely at locations in the image which are the most likely to contain edges. The sparseness is assured using a classical winner–take–all operation, ensuring that the generative patches of the primitives do not overlap. Each primitive encodes the image information contained by a local image patch. Multi–modal information is gathered from this image patch, including the

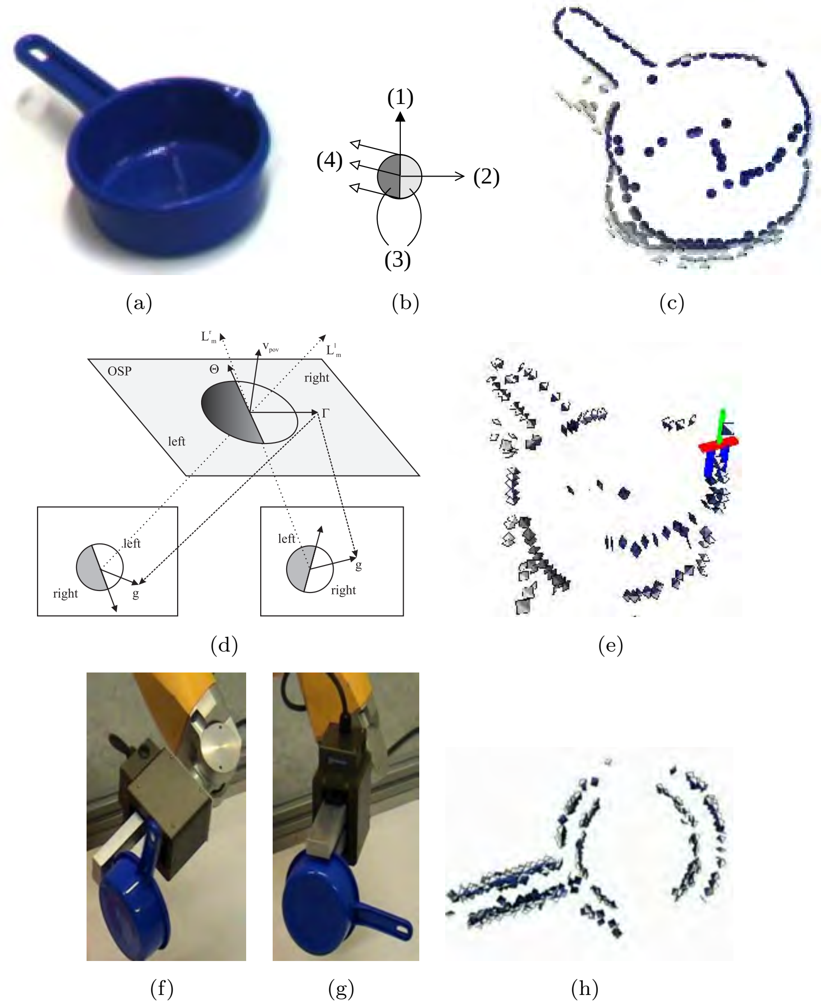


Fig. 1. Overview of the system. (a) Image of the scene as viewed by the left camera at the first frame. (b) Symbolic representation of a primitive wherein (1) shows the orientation, (2) the phase, (3) the colour, (4) the optic flow of the primitive. (c) 2D primitives extracted at one object in the scene from (a). (d) Illustration of the reconstruction of a 3D primitive from a stereo pair of 2D primitives. (e) 3D primitives reconstructed from the scene and one grasping hypothesis. (f)-(g) Two views of robot rotating the grasped object to build its 3D representation. (h) The learned 3D representation of the object.

position $\boldsymbol{x}$ of the center of the patch, the orientation θ of the edge, the phase ω of the signal at this point, the colour $\boldsymbol{c}$ sampled over the image patch on both sides of the edge, the local optical flow $\boldsymbol{f}$ and the size of the patch ρ. Consequently a local image patch is described by the following multi–modal vector:

$$\pi = (\boldsymbol{x}, \theta, \omega, \boldsymbol{c}, \boldsymbol{f}, \rho)^T, \tag{1}$$

that we will name *2D primitive* in the following. The primitive extraction process is illustrated in figure 1.

In a stereo scenario, *3D primitives* can be computed from correspondences of 2D primitives (figure 1)

$$\Pi = (\boldsymbol{X}, \boldsymbol{\Theta}, \Omega, \boldsymbol{C})^T, \tag{2}$$

where $\boldsymbol{X}$ is the position in space, $\boldsymbol{\Theta}$ is the 3D orientation, Ω is the phase of the contour, and $\boldsymbol{C}$ is the colour on both sides of the contour.

2.2. *Perceptual relations between primitives*

The sparseness of the primitives allows for the formulation of four *structural relations* between primitives that are crucial in our context since they allow us to relate feature constellations to grasping actions (in the first OAC in section 3) or visual percepts in consecutive frames (in the second OAC described in section 4). See Kalkan et al.[15] for more details.

Co–planarity Two spatial primitives Π_i and Π_j are co–planar iff their orientation vectors lie on the same plane. The co–planarity relation is illustrated in Fig. 2(b). In the context of the grasping reflex described in section 3 grasping actions become associated to the plane spanned by co-planar primitives.

Collinear grouping (i.e., collinearity): Two 3D primitives Π_i and Π_j are collinear (i.e., part of the same group) iff they are part of the same contour. Due to uncertainty in 3D reconstruction process, in this work, the collinearity of two spatial primitives Π_i and Π_j is computed using their 2D projections π_i and π_j. Collinearity of two primitives is illustrated in Fig. 2(a).

Co–colority: Two spatial primitives Π_i and Π_j are co–color iff their parts that face each other have the same color. In the same way as collinearity, co–colority of two spatial primitives Π_i and Π_j is computed using their 2D projections π_i and π_j. Fig. 2(c), a pair of co–color and non co–color primitives are shown.

Testing for collinearity and co–colority help to reduce the number of generated grasping hypotheses (see section 3.2).

Rigid body motion: The change of position and orientation induced by a rigid body motion between two frames at time t and $t+1$ ($\Pi^{t+1} = \mathrm{RBM}(\Pi^t)$) can be computed analytically[16], phase and colour can be approximated to be constant.

3. Grasping Reflex

In this section, we describe the first OAC that leads to a physical control over objects. Note that a high success rate is not important in this context, but more that the success can be evaluated by haptic feedback which then gives indications to proceed with another OAC described in section 4.

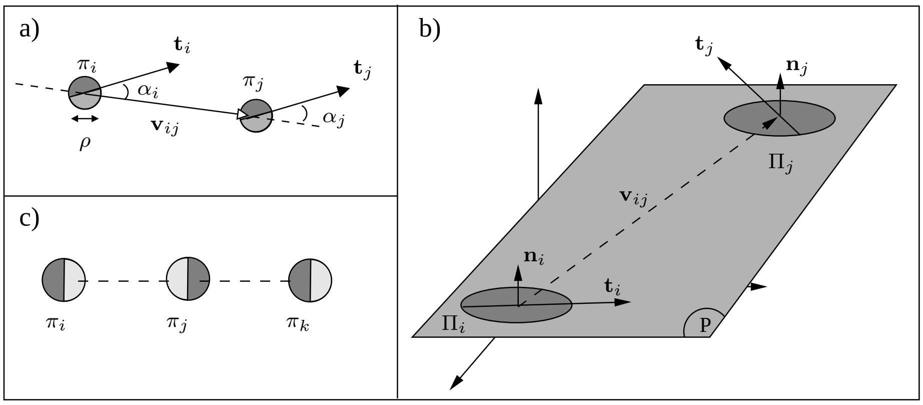


Fig. 2. Illustration of the relations between a pair of primitives. **(a)** Collinearity of two 2D primitives π_i and π_j. **(b)** Co–planarity of two 3D primitives Π_i and Π_j. **(c)** Co–colority of three 2D primitives π_i, π_j and π_k. In this case, π_i and π_j are cocolor, so are π_i and π_k; however, π_j and π_k are not cocolor.

3.1. *Elementary grasping actions associated to co-planar Primitives*

Coplanar relationships between visual primitives suggests different graspable planes. Fig. 3(a) shows a set of spatial primitives on two different contours l_i and l_j with co–planarity, co–colority and collinearity relations.

Four elementary grasping actions (EGA) will be considered as shown in Fig. 3,b–e. EGA1 is a "pinch" grasp on a thin edge like structure with approach direction along the surface normal of the plane spanned by the primitives. EGA2 is an "inverted" grasp using the inside of two edges with approach along the surface normal. EGA3 is a "pinch" grasp on a single edge with approach direction perpendicular to the surface normal. EGA4 is wide grasp making contact on two separate edges with approach direction along the surface normal.

The EGAs will be parameterized by their final pose (position and orientation) and the initial gripper configuration. For the simple parallel jaw gripper, an EGA will thus be defined by seven parameters: EGA$(x, y, z, k, l, m, \delta)$ where $\mathbf{p} = [x, y, z]$ is the position of the gripper "center" according to Fig. 3(f); k, l, m are the roll, pitch and yaw angles of the vector $\mathbf{n}$; and δ is the gripper configuration, see Fig. 3(f). Note that the gripper "center" is placed in the "middle" of the gripper.

We intend to compute these grasp parameters from coplanar pairs of 3D–primitives. Let $\Gamma = \{\Pi_1, \Pi_2\}$ be a primitive pair for which the coplanar relationship is fulfilled. Let Γ_i be the ith pair and $\mathbf{p}$ the plane defined by the coplanar relationship of the primitives of Γ_i. Let $\Lambda(\Pi)$ be the position of Π and $\Theta(\Pi)$ be the orientation of Π. The parameterization of the EGAs is given with the gripper normal $\mathbf{n}$ and the normal $\mathbf{a}$ of the surface between the two fingers as illustrated in Fig. 3(f). From this, the yaw, pitch and roll angles can be easily computed. For

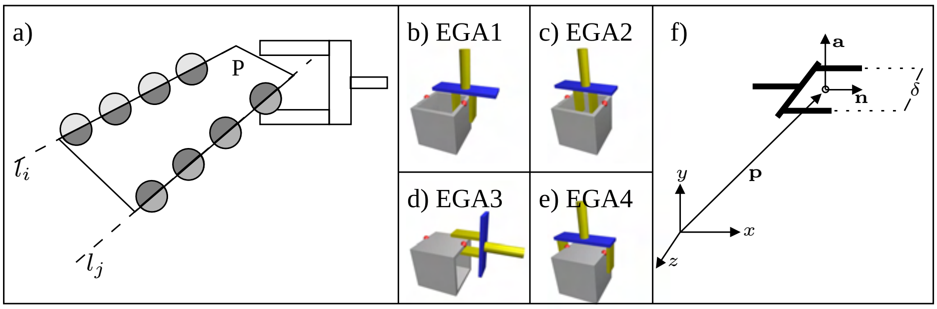


Fig. 3. **(a)** A set of spatial primitives on two different contours l_i and l_j that have co–planarity, co–colority and collinearity relations; a plane P defined by the co–planarity of the spatial primitives and and an example grasp suggested by the plane. **(b)-(e)** Elementary grasping actions, EGA1, EGA2, EGA3 and EGA4 respectively. **(f)** Parameterization of EGAs.

example for EGA1, there will be two possible parameter sets given the primitive pair $\Gamma = \{\Pi_1, \Pi_2\}$. The parameterization is as follows:

$$\begin{aligned} \mathbf{p}_{\text{gripper}} &= \Lambda(\Pi_i), \\ \mathbf{n} &= \nabla(\mathbf{p}), \\ \mathbf{a} &= \mathbf{perp_n}(\Theta(\Pi_i)) / \parallel \mathbf{perp_n}(\Theta(\Pi_i)) \parallel \text{ for } i = 1, 2, \end{aligned} \tag{3}$$

where $\nabla(\mathbf{p})$ is the normal of the plane $\mathbf{p}$ and $\mathbf{perp_u}(\mathbf{a})$ is the projection of $\mathbf{a}$ perpendicular to $\mathbf{u}$.

The main motivation for choosing these grasps is that they represent the simplest possible two fingered grasps humans commonly use which can also be simulated on our robot system. The result of applying the EGAs can be evaluated by the information given by the gripper (Schunk, PT–AP 70) which gives the distance between the two jaws at each instance of time.

For EGA1, EGA3 and EGA4, a failed grasp can be detected by the fact that the gripper is completely closed. For EGA1 and EGA3, the expected grasp is a pinch type grasp, i.e. narrow. Therefore, they can also "fail" if the gripper comes to a halt too early. EGA2 fails if the gripper is fully opened, meaning that no contact was made with the object. If none of the above situations is encountered the EGA is considered successful. The details of how EGAs are computed can be found in Aarno et al.[11].

3.2. *Limiting the number of actions*

For a typical scene, the number of coplanar pairs of primitives is on the order of $10^3 - 10^4$. Given that each coplanar relationship gives rise to six different grasps from the four different categories, it is obvious that the number of suggested actions must be further constrained. In addition, there exist many coplanar pairs of primitives affording similar grasps.

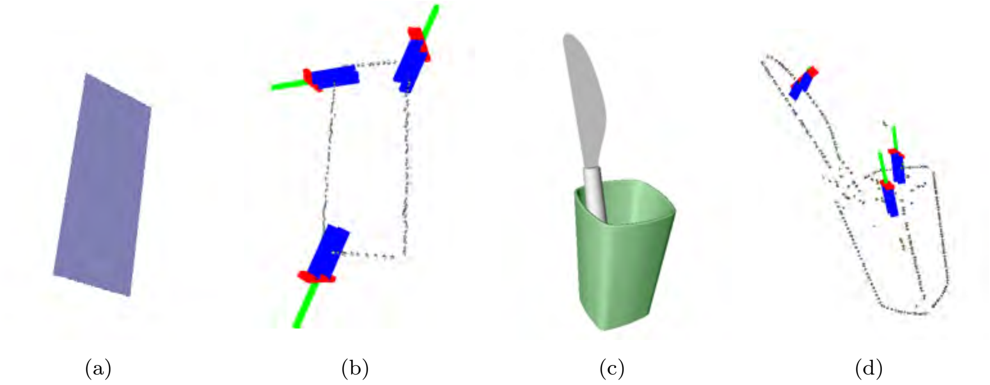


Fig. 4. Two example scenes designed for testing **(a,c)** and a selection of the generated actions **(b,d)**.

To overcome some of the above problems, we make use of the structural richness of the primitives. First, their embedding into collinear groups naturally clusters the grasping hypotheses into sets of redundant grasps from which only one needs to be tested. Furthermore, co–colority, gives an additional hypothesis for a potential grasp. Aarno et al.[11] quantified the reduction in EGAs hypotheses using collinearity and co–colority in a simulation environment, showing that the number of EGAs can be reduced systematically.

3.3. *Experimental evaluation*

To evaluate the grasping reflex we made experiments within the simulation environment GraspIt[17] and with a real scene. In the GraspIt environment, we evaluated success rate on scenes of different complexity (see Fig. 4 for a number of successful grasps on two scenes). Success rate was dependent on the scene complexity, ranging from appr. 90%[c] in the case of a simple plane (see Fig. 4a,b), to around 25% for scenes of larger complexity 4c,d.

We then evaluated the exploration strategy on a real scene (see Fig. 5(a)). After reconstructing 3D–primitives from stereo images (Fig. 5(b)), 912 EGAs were generated. However, in a real set–up there are additional constraints such as the definition of a region of interest where objects are supposed to occur, the fact that not all EGAs are actually performable due to limited workspace.[d] In addition, grasps leading to collisions with the floor or the wall need to be eliminated. Table 1 shows effects of reductions.

In a full exploration sequence, the system attempts to perform one of the 50

[c] A success is counted when one of the six EGAs has been performed successfully

[d] Note that workspace needs to be defined in terms of a 6D pose and that even when a 3D point is reachable, it is not certain that the desired end–effector orientation can be achieved at this point.

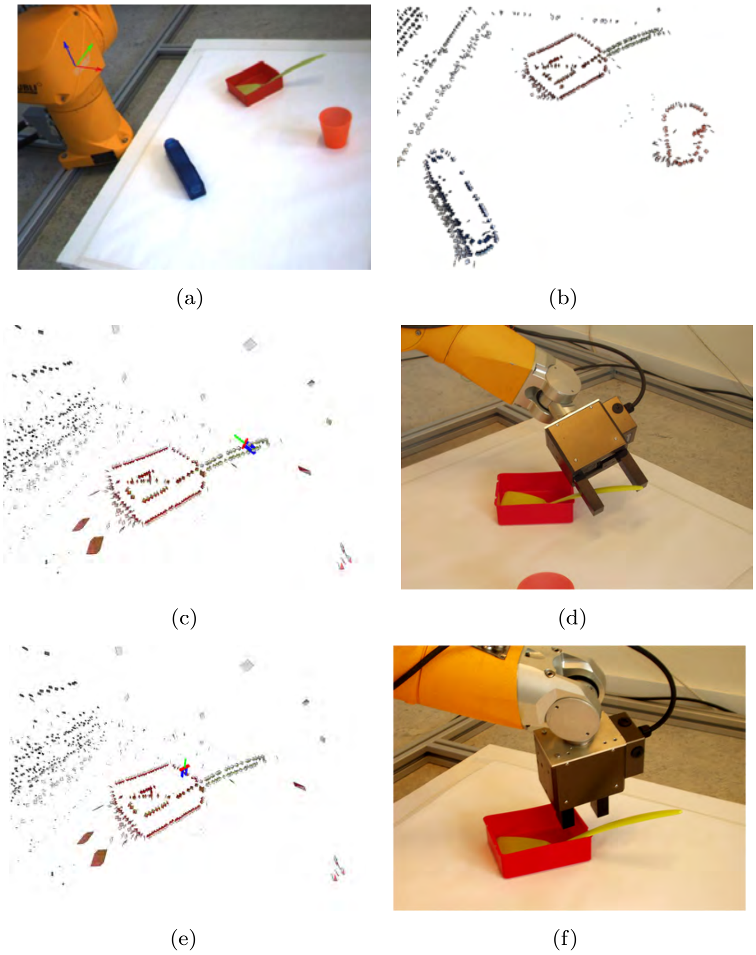


Fig. 5. The experimental scene for testing EGAs. **(a)** The view from the left camera. The origin and orientation of world coordinate frame are illustrated in top left corner. **(b)** Extracted 3D primitives displayed in the visualization environment module. **(c,e)** Successful grasps shown in our 3D displaying software. **(d,f)** The robot arm executing the grasps shown in (c,e).

remaining EGAs. A failure to grasp an object generally causes changes in the scene, and the whole sequence of capturing images, generating and reducing EGAs would be repeated. However, for the purpose of evaluating the whole set of proposed EGAs for a single scene, the objects in this experiment were placed at their original position after each attempted grasp.

In the specific scenario shown in Fig. 5, three out of the four objects could be grasped by the reflex. Out of 50 grasps, 7 lead to physical control over objects. In one case, the contact area was too small, leading to an unstable grip, and the accumulation module (see section 4.1) could not be applied.

Table 1. The results of applying reductions to the initial set of EGAs.

Reductions:	Initial number of EGAs	deleted	remaining
to region of interest	912	684	228
to reachable configurations	228	123	105
collision free (floor)	105	55	50
collision free (self)	50	0	50

4. Detection of Objectness and Object Shape

Having achieved physical control over an object, measured by the distance between the gripper's jaws after closing or opening (in case of EGA2), a second OAC is triggered that makes use of the additional capability that has the agent of actively manipulating the object.

If an object's motion within the scene is known, then the relation between this object's features in two subsequent frames becomes deterministic (excluding the usual problems of occlusion, sampling, etc.). This means that a 3D–primitive that is present in one frame is subject to a transformation that is fully determined by the object's motion: generally a change of 3D position and 3D orientation.[e] If we assume that the motion between consecutive frames is reasonably small then a contour will not appear or disappear unpredictably, but will have a life–span in the representation, between the moment it enters the field of view and the moment it leaves it. Assuming having a fully calibrated system and having physical control over the object (as gained by the first OAC described in section 3) we can compute the 3D–primitives' change in camera coordinates.

These predictions are relevant in different contexts:

Establishment of objectness: The objectness of a set of features is characterized by the fact that they all move according to the robot's motion. This property is discussed in the context of a grounded AI planning system in Geib et al.[18].

Segmentation: The system segments the object from the rest of the scene using its predicted motion.

- **Disambiguation:** Erroneous 3D–primitives can be characterized (and eliminated) by inconsistent motion according to the predictions.

Learning the object model: A full 3D model of the object can be extracted by merging different $2\frac{1}{2}$D views created by the motion of the end effector.

4.1. *Making predictions from the Robot Motion*

If we consider a 3D–primitive $\Pi_i^t \in \boldsymbol{\mathcal{S}}_t$ describing an object's contour at time instant t, and assume that the object's motion is known between the two instants t and $t + \Delta t$, then we can predict this primitive's position at $t + \Delta t$.

[e] We neglect the effects of lighting and reflection, and assume that phase and colour stay constant.

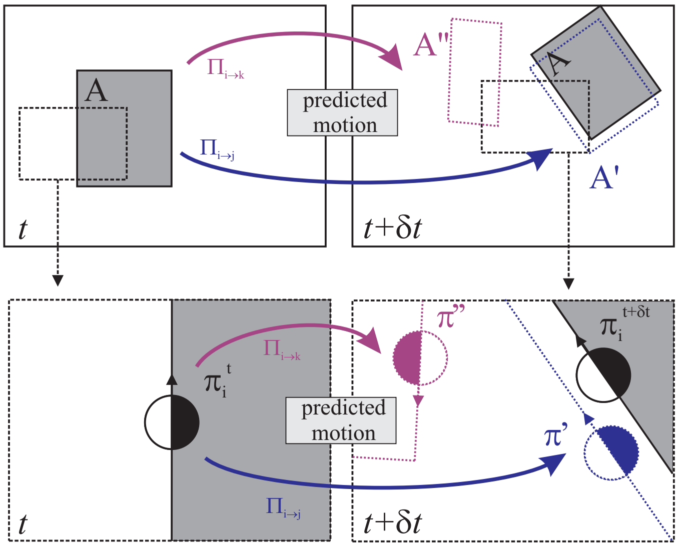


Fig. 6. Example of the accumulation of a primitive (see text) [NK: I see one problem with this figure. A rectangle has two meanings once it is a frame that becomes enlarged in the bottom image and once it is an object].

The projection of 3D–primitives to the image domain predict where 2D–primitives should be extracted from each camera's image at time $t + \Delta t$. It is then possible to assess the correctness of a reconstructed 3D–primitive by how reliably it is confirmed by subsequently extracted 2D–primitives.

This prediction/verification process is illustrated in Fig. 6. The left column shows an image from a scene at time t; the right column shows an image from the same scene, taken at a later time $t + \Delta t$. The top row shows the complete image of the object; the bottom row shows object's details indicated by the black rectangle. If we consider the object **A** (solid rectangle in the top–left and top–right images) that is subjected to a motion $M_{t\to t+\Delta t}$, between time t and $t + \Delta t$ (as illustrated in the figure's top row). Two object's shape hypotheses generate two distinct predictions at time $t + \Delta t$: **A'** (correct and similar to the actual pose of the object, denoted by the blue rectangle in the top–right image) and **A"** (erroneous, red rectangle). The consequences for individual primitives on the object is shown in the bottom row: the primitive π_i^t lies on the contour of **A** at the instant t (bottom–left image). Two plausible stereo correspondences π_j^t and π_k^t at time t, lead to the reconstruction of two mutually exclusive 3D–primitives $\Pi_{i\to j}^t$ and $\Pi_{i\to k}^t$, and thus the prediction of two different poses at time $t + \Delta t$: 1) the correct hypothesis $\Pi_{i\to j}^t$ predicts a 2D–primitive π' that matches closely with one of the a 2D–primitive $\pi_i^{t+\Delta t}$ (blue in the bottom–right image), newly extracted at $t + \Delta t$ from the contour of **A**, thus comforting the original hypothesis; 2) the incorrect hypothesis $\Pi_{i\to k}^t$ predicts a 2D–primitive π'' (red in the bottom–right image), that do not match any primitive extracted from the image, thereby revealing the erroneousness of the hypothesis.

We then propose to use these predictions to re–evaluate 3D–primitives' confidence depending on their resilience over time. This is justified by the continuity assumption, which states that 1) scene's objects or contours should not appear and disappear abruptly from the field of view (FoV) but move in and out gracefully according to the estimated ego–motion; and 2) a contour's position and orientation at any point in time is fully determined by the knowledge of its position at a previous instant in time and of its motion since.

Consider a primitive Π_i, predicting a primitive $\hat{\Pi}_i^t$ at time t. We write the fact that this prediction is confirmed by the images at time time t as $\mu_t(\hat{\Pi}_i) = 1$; and the fact that it is not confirmed (i.e., there is no 2D–primitive extracted at time t that is similar to the projection of $\hat{\Pi}_i^t$ on the image plane) as $\mu_t(\hat{\Pi}_i) = 0$. By extension, we code the resilience a primitive Π_i, from its apparition at time 0 until time t as the binary vector:

$$\boldsymbol{\mu}(\Pi_i) = \left(\mu_t(\hat{\Pi}_i), \mu_{t-1}(\hat{\Pi}_i), \cdots, \mu_0(\hat{\Pi}_i)\right)^T. \tag{4}$$

We then apply Bayes formula to evaluate the posterior likelihood that a 3D–primitive is correct knowing its resilience vector:

$$p\left(\Pi_i | \boldsymbol{\mu}(\Pi_i)\right) = \frac{p\left(\boldsymbol{\mu}(\Pi_i)|\Pi\right) p\left(\Pi\right)}{p\left(\boldsymbol{\mu}(\Pi_i)|\Pi\right) p\left(\Pi\right) + p\left(\boldsymbol{\mu}(\Pi_i)|\bar{\Pi}\right) p\left(\bar{\Pi}\right)}. \tag{5}$$

In this formula, Π and $\bar{\Pi}$ are correct and erroneous primitives, respectively. The quantities $p\left(\Pi\right)$ and $p\left(\bar{\Pi}\right)$ are the prior likelihoods for a 3D–primitive to be correct and erroneous. The quantity $p\left(\boldsymbol{\mu}(\hat{\Pi}_i)|\Pi\right)$ (resp. $p\left(\boldsymbol{\mu}(\Pi_i)|\bar{\Pi}\right)$) expresses the probability of occurrence of a resilience vector $\boldsymbol{\mu}(\Pi_i)$ for a correct (resp. erroneous) primitive Π_i.

Furthermore, if we assume independence between the matches $\mu_t(\hat{\Pi}_i)$, then for a primitive Π_i that exists since n iterations and has been matched successfully m times, we have the following relation:

$$\begin{aligned} p\left(\boldsymbol{\mu}(\hat{\Pi}_i)|\Pi\right) &= \textstyle\prod_t p\left(\mu_t(\hat{\Pi}_i)|\Pi\right) \\ &= p\left(\mu_t(\hat{\Pi}_i) = 1|\Pi\right)^m p\left(\mu_t(\hat{\Pi}_i) = 0|\Pi\right)^{n-m}. \end{aligned} \tag{6}$$

In this case the probabilities for μ_t are equiprobable for all t, and therefore if we define the quantities $\alpha = p\left(\Pi\right)$, $\beta = p\left(\mu_t(\hat{\Pi}) = 1|\Pi\right)$ and $\gamma = p\left(\mu_t(\hat{\Pi}) = 1|\bar{\Pi}\right)$ then we can rewrite Eq. (5) as follows:

$$p\left(\Pi_i|\bar{\boldsymbol{\mu}}(\hat{\Pi}_i)\right) = \frac{\beta^m(1-\beta)^{n-m}\alpha}{\beta^m(1-\beta)^{n-m}\alpha + \gamma^m(1-\gamma)^{n-m}(1-\alpha)}. \tag{7}$$

We measured these prior and conditional probabilities using a video sequence with known motion and depth ground truth obtained via range scanner. We found values of $\alpha = 0.46$, $\beta = 0.83$ and $\gamma = 0.41$. This means that, in these examples, the prior likelihood for a stereo hypothesis to be correct is 46%, the likelihood for a correct hypothesis to be confirmed is 83% whereas for an erroneous hypothesis it is of 41%. These probabilities show that Bayesian inference can be used to identify

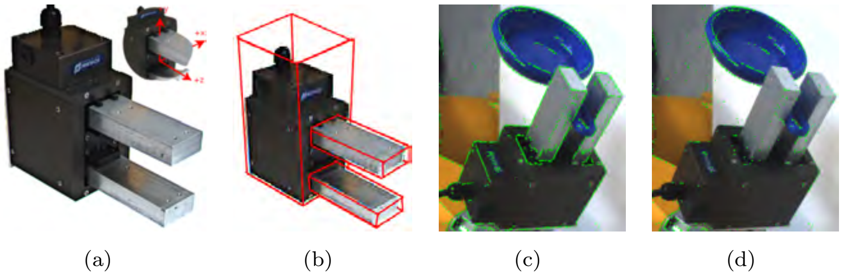


Fig. 7. Gripper elimination (a) grasper and grasper coordinate system (b) bounding boxes of grasper body and its fingers (c) primitives before grasper elimination (d) primitives after grasper elimination

correct correspondences from erroneous ones. To stabilize the process, we will only consider the n first frames after the appearance of a new 3D–primitive. After n frames, the confidence is fixed for good. If the confidence is deemed too low at this stage, the primitive is forgotten. During our experiments $n = 5$ proved to be a suitable value.

The end-effector of the robot follows the same motion as the object. Therefore, this end-effector becomes extracted as well. Since we know the geometry of this end-effector (Fig. 7 (a)), we can however easily subtract it by eliminating the 3D primitives that are inside the bounding boxes that bounds the body of the gripper and its fingers (Fig. 7 (b)). For this operation, three bounding boxes are calculated in grasper coordinate system. In Fig. 7 (c) 2D projection of 3D primitives extracted from a stereo pair is presented. After gripper elimination, the 2D projections of remaining primitives are shown in Fig. 7 (d).

4.2. *Experiments*

We applied the accumulation scheme to a variety of scenes where the robot arm manipulated several objects. The motion was a rotation of 5 degrees per frame. The accumulation process on one such object is illustrated in Fig. 8. The top row show the predictions at each frame. The bottom row, shows the 3D–primitives that were accumulated (frames 1, 12, 22, and 32). The object representation becomes fuller over time, whereas the primitives reconstructed from other parts of the scene are discarded. Fig. 9 shows the accumulated representation for various objects. The hole in the model corresponds to the part of the object occluded by the gripper. Accumulating the representation over several distinct grasps of the objects would yield a complete representation.

5. Conclusion

We introduced a scheme in which two modules in terms of Object Action Complexes (OACs) become combined to extract world knowledge in terms of the objectness

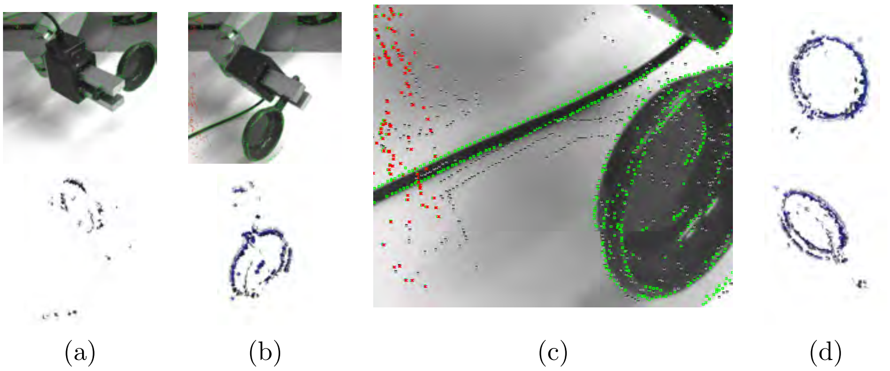


Fig. 8. Birth of an object (a)-(b) top:2D projection of the accumulated 3D representation and newly introduced primitives, bottom:accumulated 3D representation. (c) newly introduced and accumulated primitives in detailed. Note that, the primitives that are not updated are red and the ones that have low confidence are grey (d) final accumulated 3D representation from two different poses.

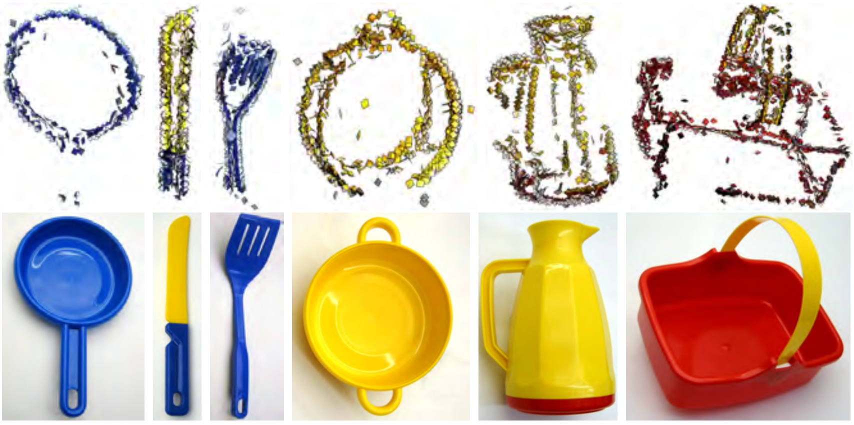


Fig. 9. Objects and their related accumulated representation.

of a set of local features as well as the object shape. Although this exploration scheme is completely autonomous, we argued that there is a significant amount of prior knowledge in terms of generic properties of the world built into the system. Starting with a rather sophisticated feature extraction process covering common visual modalities, functional relations defined on those features such as co-planarity, co-linearity, basic laws about Euclidean geometry and the motion of rigid object has been exploited. Furthermore and at least of equal importance, it was the capability to act on the world that made this process possible. Here the embodiment of the agent is of high importance. The option to grasp and move the objects in a controlled way is rather unique to few species and with high likelihood linked to develop higher cognitive capabilities.

The work described in this paper is part of the EU project PACOplus[19] which aims at a system covering different levels of cognitive processing from low-level processes as described here up to a planning AI level (see Geib et al[18]). This work introduced describes an important module of such a cognitive system which gives information that higher levels require to start operating. First, it segments the world in objects which are the basic entities that higher level reasoning is based on. Moreover, it generates 3D object representations in a procedural way which then can be used for object identification and pose estimation (see, e.g., Lowe[2] for the use of 3D models for object recognition and Detry and Piater[20] for first steps in directly making use of the extracted representations described in this paper). By the described exploratory procedure, a natural mechanism is given that enlarges the internal world model that then can be used by higher levels for reasoning and planning (for first steps, see Geib et al[18]).

Acknowledgements: We would like to thank Tamim Asfour, Mark Steedman, Christopher Geib and Ron Petrick for fruitful discussions. This work has been funded by the EU project PACOPlus[19].

References

1. J. Gibson, *The Ecological Approach to Visual Perception* (Boston, MA: Houghton Mifflin, 1979).
2. D. Lowe, Fitting parametrized 3D–models to images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**(5), 441–450 (1991).
3. N. Krüger, N. Pugeault and F. Wörgötter, Multi-modal primitives: Local, condensed, and semantically rich visual descriptors and the formalization of contextual information (also available as technical report (2007-4) of the Robotics Group Maersk Institute, University of Southern Denmark) (submitted).
4. Y. Aloimonos, I. Weiss and A. Bandopadhay, Active vision, *International journal of computer vision* **2**, 333–356 (1987).
5. R. Rao and D. Ballard, An active vision architecture based on iconic representations, *Artificial Intelligence Journal* **78**, 461–505 (1995).
6. P. Fitzpatrick and G. Metta, Grounding Vision Through Experimental Manipulation, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **361**, 2165 – 2185 (2003).
7. L. Natale, F. Orabona, G. Metta and G. Sandini, Exploring the world through grasping: A developmental approach, *CIRA 2005. Proceedings. 2005 IEEE International Symposium on Computational Intelligence in Robotics and Automation, 2005* , 559–565 (2005).
8. J. Modayil and B. Kuipers, Bootstrap learning for object discovery, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-04)* , 742–747. (2004).
9. P. Kellman and M. Arterberry (eds.), *The Cradle of Knowledge* (MIT-Press, 1998).
10. N. Krüger and F. Wörgötter, Statistical and deterministic regularities: Utilisation of motion and grouping in biological and artificial visual systems, *Advances in Imaging and Electron Physics* **131**, 82–147 (2004).
11. D. Aarno, J. Sommerfeld, D. Kragic, N. Pugeault, S. Kalkan, F. Wörgötter, D. Kraft and N. Krüger, Early reactive grasping with second order 3d feature relations, *IEEE*

International Conference on Robotics and Automation (ICRA), Workshop: From features to actions - Unifying perspectives in computational and robot vision (2007).

12. N. Pugeault, E. Baseski, D. Kraft, F. Wörgötter and N. Krüger, Extraction of multi–modal object representations in a robot vision system (2007).
13. N. Krüger, M. Lappe and F. Wörgötter, Biologically motivated multi-modal processing of visual primitives, *Interdisciplinary Journal of Artificial Intelligence & the Simulation of Behaviour, AISB Journal* **1**(5), 417–427 (2004).
14. J. H. Elder, Are edges incomplete?, *International Journal of Computer Vision* **34**, 97–122 (1999).
15. S. Kalkan, N. Pugeault and N. Krüger, Perceptual operations and relations between 2D or 3D visual entities, (2007-3) (2007).
16. O. Faugeras, *Three–Dimensional Computer Vision* (MIT Press, 1993).
17. A. Miller and P. K. Allen, Graspit!: A versatile simulator for robotic grasping, *IEEE Robotics and Automation Magazine* **11**(4), 110–122 (2004).
18. 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, *Workshop 'Toward Cognitive Humanoid Robots' at IEEE-RAS International Conference on Humanoid Robots (Humanoids 2006)* (2006).
19. Pacoplus: Perception, action and cognition through learning of object-action complexes, *IST-FP6-IP-027657, Integrated Project* (2006-2010).
20. R. Detry and J. Piater, Hierarchical integration of local 3d features for probabilistic pose recovery (2007).