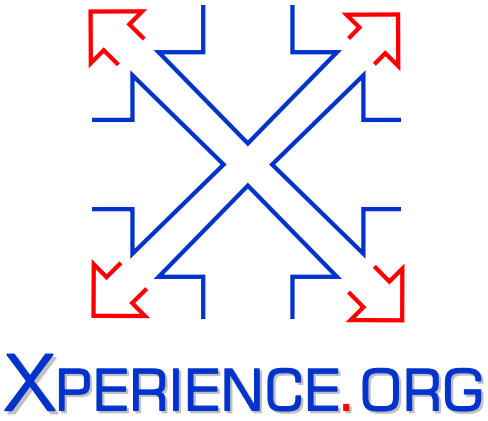
Project Acronym:	Xperience
Project Type:	IP
Project Title:	Robots Bootstrapped through Learning from Experience
Contract Number:	270273
Starting Date:	01-01-2011
Ending Date:	31-12-2015



Deliverable Number:	D3.1.1
Deliverable Title:	Structural Bootstrapping on Sensorimotor Experience (I): Report or scientific publication on construction and application of the generative inner model
Type (Internal, Restricted, Public):	PU
Authors:	J. Piater, S. Szedmak, F. Wörgötter, E. Aksoy, J. Papon, M. Tamosiunaite, D. Kraft, N. Krüger, A. Ude
Contributing Partners:	UIBK, UGOE, SDU, JSI

Contractual Date of Delivery to the EC:	31-01-2013
Actual Date of Delivery to the EC:	31-01-2013

Contents

Chapter 1

Executive Summary

This document summarizes the efforts of the consortium in developing methods for generalization from sensorimotor experience, using statistical methods as well as structural bootstrapping, including the formation of object and action categories based on their properties experienced. These contributions were generally created under WP3.1.

1.1 General Objective of WP3.1: Structural Bootstrapping on sensorimotor experience

The ultimate goal is to learn cognitive categories faster than with existing methods, as well as to learn more elaborate cognitive categories than can feasibly be done with existing methods.

We draw on two key concepts to achieve this:

- First, generative models are constructed on the basis of **accumulated experience**.
- Secondly, **accumulated experience will be mined to erect scaffolding** for future learning problems, constraining them to accelerate learning and to learn more from less real-world interaction.

Expected Outcome: **WP3.1 will produce learning methods that can learn relatively elaborate sensorimotor concepts relatively efficiently by informing future learning problems using past experience.**

1.2 An Economic View of the Objective

The objective can be stated in a quantifiable form as:

Based on the accumulated knowledge collected in the experiments, efficiently reduce the cost (time, energy, human interaction) in the robot learning cycle.

The cost reduction and its efficiency can be measured. Therefore it can be proved that our methods can be used at smaller cost and in a broader range of applications than other available methods.

We expect to produce first results that quantify such accelerated learning by the end of Year 3.

1.3 Outline of this Document

Sensorimotor structural bootstrapping as we pursue it in Xperience requires a fairly complex interplay of various functionalities, including transfer of learned action parameters to new, related actions, recognizing action sequences as instances of known or novel behaviors, categorizing objects based on shape as well

as on actions they afford, associating actions and their parameters to objects, parts and features, and so on.

This document begins with an interdisciplinary study of the emergence of tool use in human infants, using psychological experiments and computational models (Chapter 2). The study illuminates how repeated manipulation of objects and observation of its effects on other objects leads to differentiated, goal-directed manipulations of the former objects (tools) to achieve intended effects on the latter objects (targets). This is a concrete example of infants learning manipulations efficiently following prior, exploratory interactions, i.e., a type of sensorimotor structural bootstrapping. Although not addressing directly the core mechanisms of structural bootstrapping, the study is conceptually of immediate interest to Xperience, since it describes the interplay between the development of increasingly efficient behavior and internal representations.

The following chapters 3 through 4 present some core technical achievements that will become part of Xperience structural-bootstrapping systems.

Chapter 3 shows how motor actions can be learned efficiently on the basis of known classes of similar actions. The idea is to learn statistical models of the parameters of known actions, which typically live in low-dimensional manifolds of the entire action parameter space, and then use these models to constrain or bias future learning problems, allowing the adaptation of known actions to new actions, and greatly simplifiying the learning of new, similar actions.

Chapter 4 presents methods for assimilating action sequences given in the form of Semantic Event Chains into models of known action sequences, as well as for accommodating them as entirely new action sequences.

Chapter 5 presents an integrated learning system to achieve structural bootstrapping at the level of objects and actions. Although this takes place at a sub-symbolic and non-syntactic level, we refer to it by a new term *object-action structural bootstrapping* to differentiate it from lower-level learning in continuous sensorimotor domains such as that described in Chapter 3. Object-action structural bootstrapping is about associating action parameters to objects, transferring them to new objects, and making inferences about entire object categories based on limited interaction with objects, facilitated by previously-acquired object and action knowledge. The framework presented in Chapter 5 is intended to draw together much of the other work in perception, action and sensorimotor learning, including work from the two preceding chapters, and work from other work packages.

Chapter 2

Developmental learning of tool-use competences by means of Repetition and Differentiation

In this chapter, we describe our work on structural bootstrapping on a behavioural level. As outlined in [12], we have identified six mechanisms (repetition (M1), variation and selection (M2), differentiation (M3), decomposition (M4), composition (M5), modularisation (M6)) that guide the development of tool-use abilities in infants. These six mechanisms are supplemented by a seventh mechanism (representational re-description (M7))—crucially linked to structural bootstrapping (see figure 2.1). The diagram shows two parallel tracks of development. On the bottom is the "concrete" track which shows the development of sensorimotor schemas, which are observable in infant behaviour. On the top is the "abstract" track which shows the parallel development of the underlying representations which the infant uses.

The mechanisms M1–M6 are responsible for creating the data that delivers the empirical material for structural bootstrapping. Moreover, they interact in a complex way with the already established internal representations. Representation re-description reorganises the already accumulated knowledge.

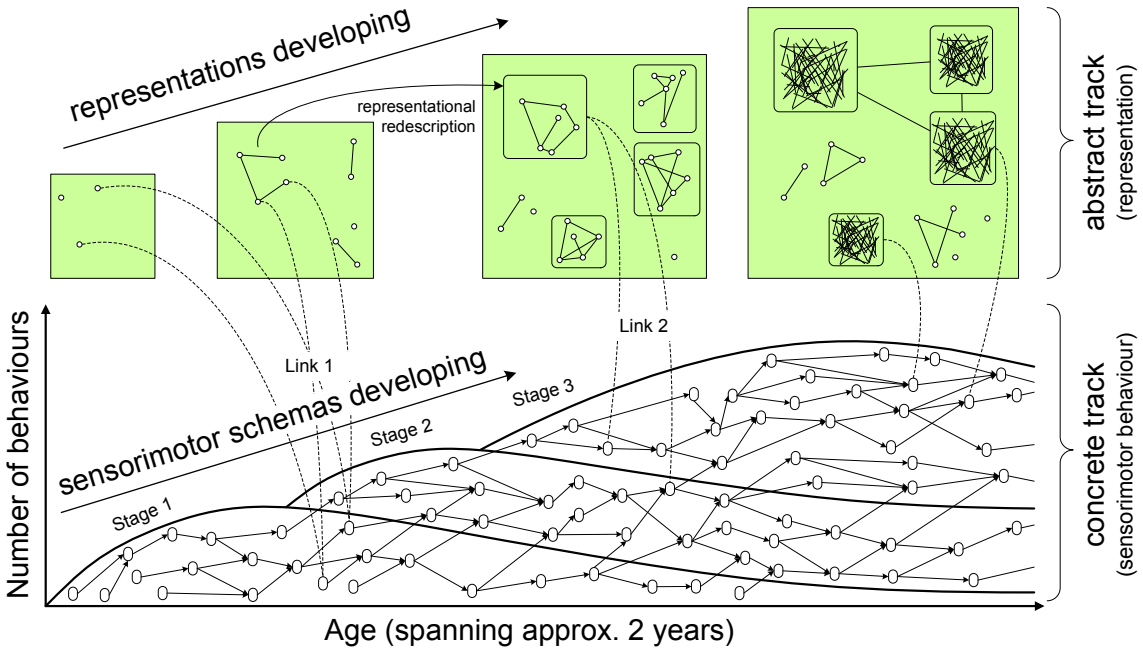


Figure 2.1: Conceptual diagram, overviewing infant developments leading to tool use; for explanation see text (taken from [12]).

In our work, we want to model the developmental process by formalising the above mentioned seven mechanisms. In the work presented here – detailed in the two attachments [FAG+12, FAK+] – we

focused on the application of the two mechanisms repetition (M1) and differentiation (M3) to learning preconditions for means-ends actions.

Our behaviour representation (called schemas in our work) consists of a tuple of *precondition*, *motor program* and *postconditions* or *goal*. In this work, we focused on learning preconditions while the motor program and the postconditions were left unchanged. As our application area, we choose means-ends behaviours (i.e. where one action is used in order to facilitate another). Figure 2.2 (top-right) shows an example for such a means-ends behaviour; by pulling the cloth the baby is able to bring the keys placed on the cloth into grasping range.

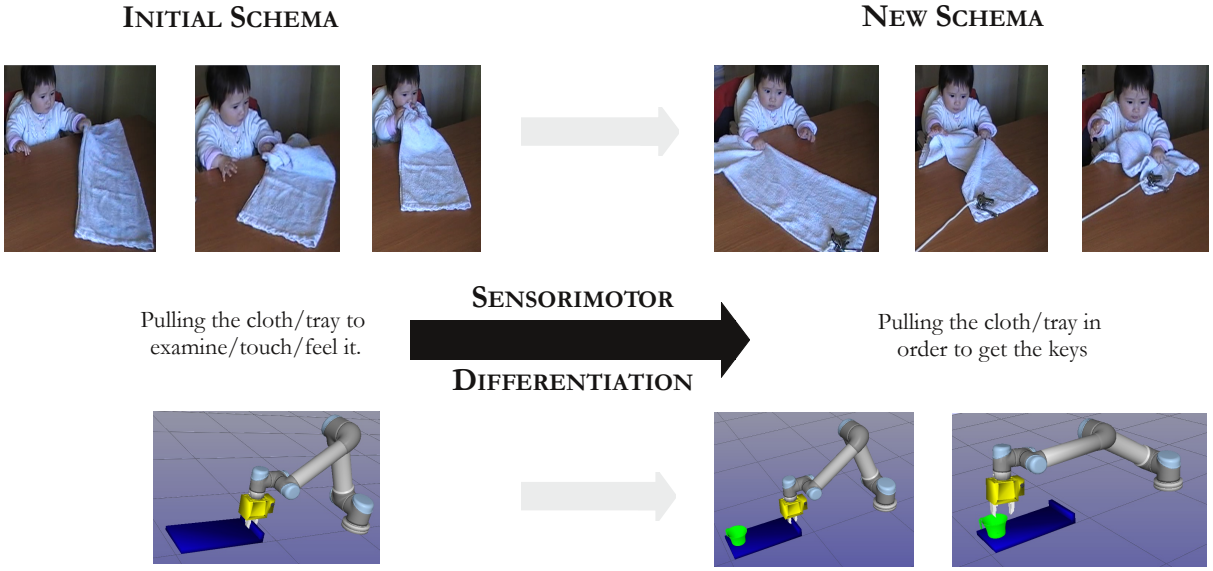


Figure 2.2: This figure illustrates how a new behaviour can be acquired by the differentiation mechanism. The original behaviour (initial schema) simply pulls a cloth/tray in order to bring it close. The new behaviour pulls the cloth/tray in order to bring the item supported by it closer. The new schema acquired will need to adjust its motor behaviour, and also to learn the situations in which this new behaviour can be expected to work. Taken from [FAK+].

The experiments in this work were performed using the dynamic simulator RobWorkStudio. This allowed us to evaluate different learning mechanisms while retaining a certain realism. Figure 2.2 bottom shows the simulation environment. To evaluate the influence of the sensor noise, we performed experiments with both (1) perfect information derived directly from the simulation environment and with (2) information derived by using an artificial RGD-D camera in the simulation environment and processing these images using an appropriate computer vision approach.

The two cases investigated are the already mentioned support scenario (key on cloth) and an obstruction scenario where one object in the scene is obstructing another which becomes graspable once the first object has been removed. The support scenario was replicated in the simulation environment by having a cup that was (or was not) placed on a plate that the robot was able to pull. For the obstruction scenario we used a cereal box object to obstruct the reach to the cup (while we made sure that the cup stayed visible for the camera).

To learn the preconditions for the individual actions (e.g., grasp cup, pull support), we employed repetition (M1). We performed a number of experiments executing these actions successfully and non successfully. Based on the situation before applying the action and the outcome of these we are able to train artificial neural networks to predict if for a given situation (a specific configuration of objects, their location in the scene and a given arm position) the motor program would be successful. In the grasp cup case, we could achieve a classification success rate of 99.16%, please see Figure 2.3 for further results.

Once we have established these actions (e.g. grasp cup and pull support) repeated applications will lead to situations where for the system unexpected things happen (e.g., the cup becomes reachable after pulling the support). In this case, we can differentiate (M3) the pull support schema into two schemas (one bringing the cup in range, the other not influencing any other objects besides the support). We have discussed a mechanisms to discover this situation in [8]. Once we have differentiated the schema

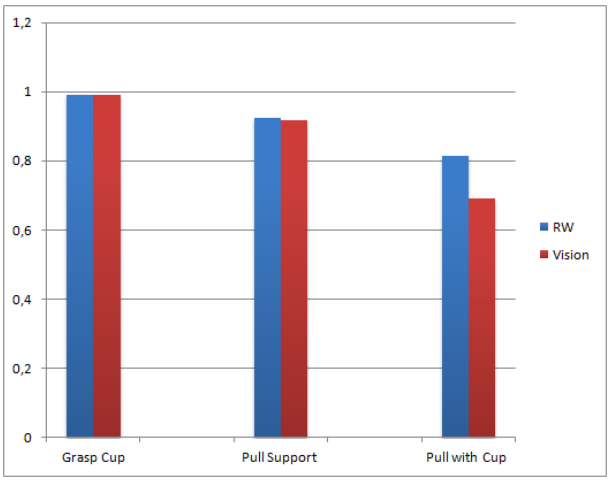


Figure 2.3: The Graph shows the accuracies of the preconditions for the three schemas Grasp Cup, Pull Support and bringing the cup into reach. It can be seen that vision based learning performs almost as well as the case were information is gained directly from the simulator (RW). 3000 training samples were used. Taken from [FAK$^+$].

into two new ones we need to adjust their motor programs (not discussed here) and their preconditions. Having learned these preconditions, we are able to predict when we can employ the means-ends action and thereby to plan action sequences that were not possible before or at least to learn to generate those plans faster than it has been possible before.

One again we used repetition (M1) and performed multiple tries with situations where the cup was placed on top of the support and where it was not. By employing the same neural network approach we were able to predict the successful applicability of the pull support with cup means-ends action in more than 80% of the situations (see Figure 2.3).

To increase the learning speed, we evaluated different learning approaches. Our focus here was on trying to choose the schema where the neural network used to predict a schema's applicability (the precondition) was least conclusive. The detailed results can be seen in [FAK$^+$]. The experiments showed that this approach works successfully in the cases where input directly from the simulator is used while it was not as successful in case of using the visually computed inputs. We hypothesise that the higher noise in the second case leads to the system selecting noisy samples rather than those that give more insight into how to classify.

Although this work does not address the core mechanisms of structural bootstrapping, it gives an important macro perspective on the developmental processes which structural bootstrapping contributes to.

Chapter 3

Accelerated sensorimotor learning in constrained domains

We use the concept of structural bootstrapping at the sensorimotor level to develop a new paradigm that combines ideas from statistical learning and reinforcement learning to efficiently compute new robot control policies. We start by collecting data from several robot executions of the desired task in different configurations of the external world. Statistical learning techniques are applied to compute a low-dimensional approximation for the optimal manifold of robot trajectories that solve the desired task in all possible configurations of the external world. The dimensionality of the approximating manifold is usually much lower than the dimensionality of the space of all robot movements. This low-dimensional manifold can be used to constrain the exploratory learning of optimal control policies for new task configurations. New control policies are obtained by means of reinforcement learning on the approximating manifold, which is much faster than learning in the full space of robot movements due to the low dimensionality of the parameter space.

We propose a reinforcement learning algorithm with an extended parameter set that combines learning in constrained domain with learning in the full space of parametric movement primitives. This way the robot can also explore actions outside of the initial approximating manifold without needing to solve any global, large-scale optimization problems. Newly obtained control policies can be used to refine the approximation for the manifold of optimal trajectories that solve the desired task. The proposed approach was tested for learning various tasks on different robots.

For more details see the attached publication [NFV+12].

Chapter 4

Assimilation and accommodation of sensorimotor experience using Semantic Event Chains

Here we present a method of expanding the robot's sensorimotor experience based on Semantic Event Chains [2] (SECs). To summarize: a SEC is a sequence of scene graphs obtained from a video sequence, where nodes represent objects and edges show the proximity relation (touching) of the objects. The SEC includes only the graphs of those frames (called key frames) where abrupt changes happen: new edges appear or edges disappear, or new nodes (objects) appear. We also attach additional information to the key frames, which includes object identity, object poses, as well as trajectories leading from one key frame to the next. Let us assume that the robot already has a set of memorized actions in the form of Semantic Event Chains populated with the additional information: objects, poses, trajectories. When a new action demonstration is observed by the robot, first a SEC is constructed from observation. Then the SEC is compared to the SECs in memory. If the new action is similar up to some threshold to another action in the memory then assimilation – memorizing a slightly modified version of the action – is performed. When there is no similar action in memory then accommodation (memorizing the action as entirely new) is performed. The framework will be described in some detail next and the full details are given in the attached manuscript [ATV+13]. The work was done in cooperation of partners UGOE and JSI, consulting with partner UEDIN.

The actions in the study are encoded using an Action Encoder matrix

$$A = \begin{bmatrix} \mathrm{X}_1 & [T.d]_{1,2} & \cdots & [T.d]_{1,m} \\ \mathrm{X}_2 & [T.d]_{2,2} & \cdots & [T.d]_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{X}_n & [T.d]_{n,2} & \cdots & [T.d]_{n,m} \end{bmatrix} \tag{4.1}$$

where in the first column we present the state of the world as described by a SEC, and in the further columns transitions $T_{i,j}$ are described:

$$T_{i,j} = \begin{cases} 0 & \text{if } r_{i,j} = r_{i,j-1},\ j > 1 \\ [T.\{d^1, d^2, \ldots, d^k\}]_{i,j} & \text{else} \end{cases} \tag{4.2}$$

where $r_{i,j}$ defines i-th relation between objects at the stage j of the process (the j-th key frame). Transitions $T.\{d^1, d^2, \ldots, d^k\}$ include the following descriptors:

- $d^1_{i,j} = \{o_{a_i}, o_{b_i}\}_i$ describe two object identifiers of those objects that are involved in the given event;
- $d^2_{i,j} = \{p\}_{i,j} = \{x, y, z, \alpha, \beta, \gamma\}_{i,j}$ is a set containing *relative* pose information *between* two object identifiers. The x, y, z and α, β, γ values hold corresponding translation and rotation values, respectively;

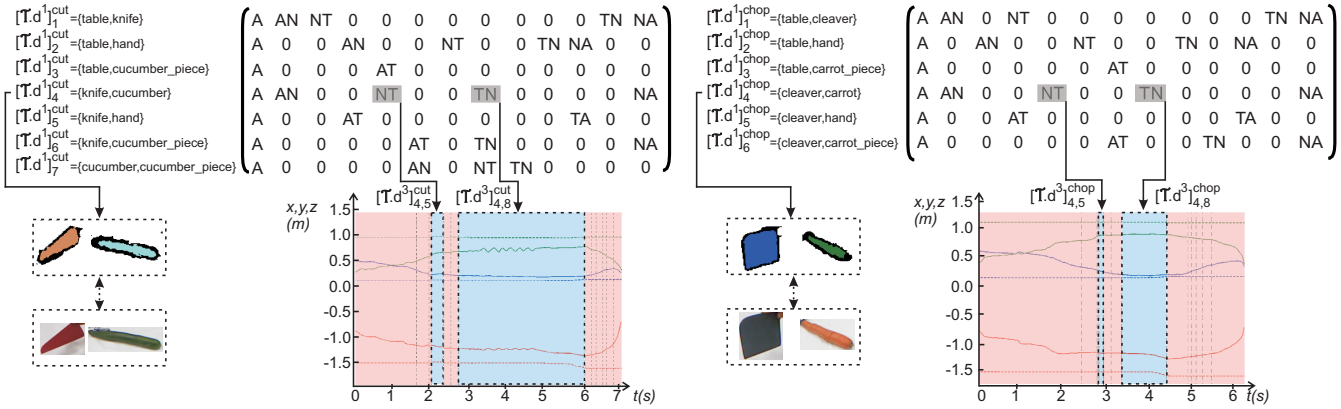


Figure 4.1: Action-encoder matrices (A) with extracted descriptors for the *Cutting* and *Chopping* actions. Movement is described in table coordinate system, x and y - table plane coordinates (red and green), z - distance from the table (blue); solid lines stand for the tool demonstrator's hand holding, dashed lines for the tip of the cucumber.

- $d^3_{i,j} = \{t\}_{i,j} = \{s, g, \tau, w_{1,\ldots,6}\}_{i,j}$, $s, g \in \mathrm{R}^6$ is a set of parameters containing trajectory information in the Cartesian task space. In this study we use the modified Dynamic Movement Primitives (DMPs, [15]).

In Figure 4.1 we show Action-encoder matrices for cutting and chopping as obtained from observation (left and right, correspondingly). Left to each matrix object identities for each line are shown. In the first matrix column, relation A means that one or both objects participating in the relation are absent from the scene. Next columns show how relations change. E.g. let us analyse the fourth line which shows the change of relations between knife and cucumber all along the action. In the second column stands a transition AN (absent-to-non-touching) which shows that the knife is appearing in the scene, but it is not touching the cucumber. Then the relations between knife and cucumber for a while stays the same, but in column five stands transition NT which shows that the relation between knife and cucumber changes from non-touching into touching. After some more non-change columns, the knife moves away from the cucumber (TN change), and so on. Beneath the Action-encoder matrix one can see images of objects involved in cutting and chopping, as well as the objects which were cut/chopped, and the trajectories, where it is shown how trajectory segments are attributed to the columns of the action encoder matrix. Due to the SEC-based trajectory segmentation one can compare corresponding trajectory pieces in two similar actions.

We have performed a comparison of three actions: cutting, chopping and stirring. We obtained that chopping is similar to cutting (both matrices in the Figure 4.1 are highly similar). Thus the chopping action was assimilated as a different variety of the cutting action. Stirring had a substantially different action descriptor matrix as compared to cutting and chopping. Thus the stirring action was accommodated as a new action. For more details on the formalism employed and similarity measures used see the attached manuscript [ATV+13].

In future stages of the project we will show how one can accumulate a bigger action library based on principles and structures described above. Note that the assimilation procedure allows us to reuse some parts of the action/skill obtained before; crucially, in this way we can obtain a non-linear speed-up. E.g. in the given chopping vs. cutting example the actions surrounding the cutting/chopping segment (everything except the part from touching the cucumber with the knife to un-touching) can be reused, as well as the pose of the knife perpendicular to cucumber with which cucumbers are most frequently cut/chopped. However, velocities and trajectories of the knife shall differ in the two varieties of the action.

Another line of current research is the improvement of the extraction of object identity and pose information based on the visual representation in the Early Cognitive Vision system described also in Section 5. In addition to simply computing a PCA on segmented scenes, a more detailed matching with object models is performed (see also [MPK13]), which should further improve the SEC representation of action sequences.

Chapter 5

Object-Action Structural Bootstrapping

Object-Action Structural Bootstrapping is about learning how to interact with objects such that (a) action parameters are associated with objects, (b) action parameters can be transferred to novel objects, and (c) entire object categories can be associated to a novel action by observing isolated examples of this novel action. Associating actions to objects (a) implicitly *categorizes* the objects by their affordances. Transferring action parameters to novel objects (b) yields efficient *skill generalization*, and the discovery of the applicability of novel actions to object categories (c) yields *object category generalization*.

The following section provides motivating examples. Subsequent sections discuss computational approaches and first experimental scenarios and results.

5.1 Motivating examples

First, imagine a robot is shown how to grasp a concave object on the inside by poking its gripper inside the concavity and expanding its fingers. Exploration shows that some objects are graspable in this way under certain object-relative gripper poses, but other objects (or inappropriate gripper poses) are not. By statistical analysis of the positive and negative grasping trials, the robot learns a classification function of object shape that tells *whether* objects are graspable-inside or not. Likewise, it learns *how* to grasp objects that are graspable-inside. This might be achieved by making the classification function a function of both shape and grasp parameters, or by learning a regression function from shape to grasp parameters.

For example, shape might be represented by relations between ECV edges and surfaces [13] as illustrated in Fig. 5.1 on the left. Here, statistical analysis identifies the fact that a configuration of two roughly parallel edges with a surface patch located between them, in a roughly parallel plane with some perpendicular offset, strongly correlates with inside-graspability using a gripper pose as illustrated.

One result of this experiment is a categorization of objects into objects that are inside-graspable and those that are not. Importantly, this categorization extends to unseen objects to the extent that the classification function generalizes appropriately.

Moreover, using the learned regression function on unseen objects is the robot's way of internally simulating the effect of imagined actions. Together, these allow the robot to generalize the learned skill of grasping inside to unseen objects.

Secondly, imagine the robot now observes a few examples of objects being dropped into a concave container, where they remain. By comparing the container with known object categories, it hypothesizes that the category of droppable-into objects coincides with the graspable-inside objects. Moreover, by comparing the action parameters, it discovers that the gripper poses relative to the concavity are identical for grasping-inside and dropping-into, except for a perpendicular offset (Fig. 5.1, right).

This allows the robot to bootstrap the new skill of dropping-into by re-using the classification function of grasping-inside, and by deriving action parameters for dropping-into from those of grasping-inside by simply applying a constant offset.

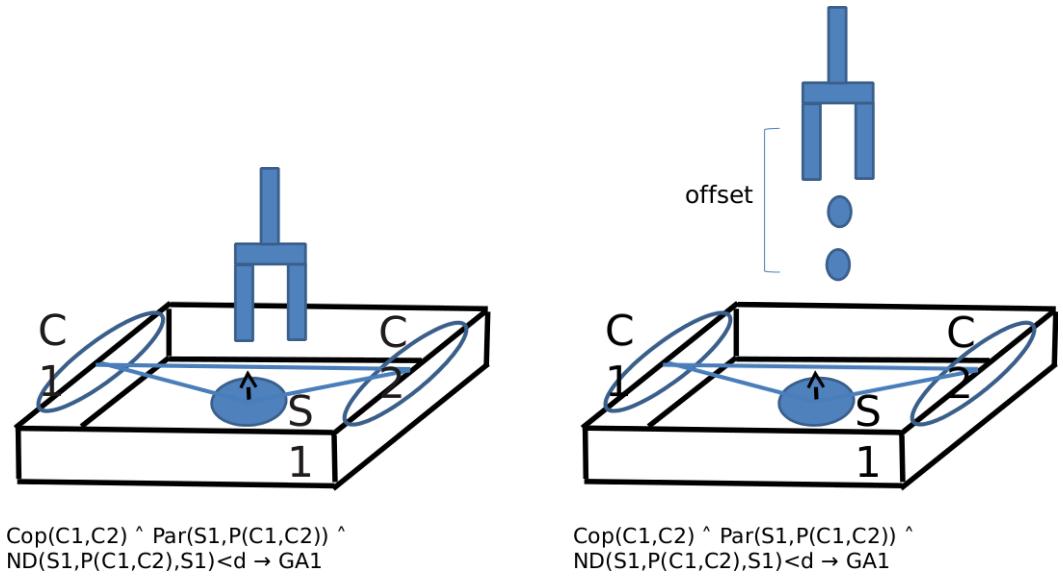


Figure 5.1: Grasping Inside (left) and Dropping Into (right).

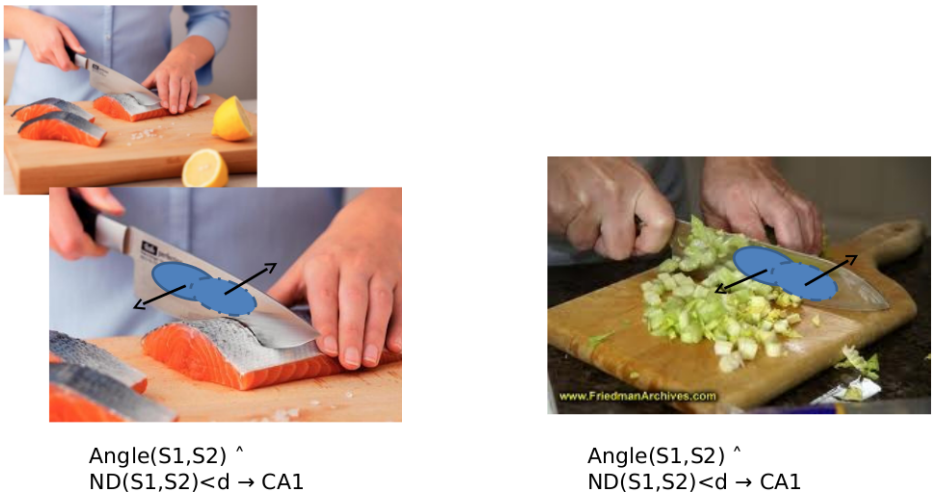


Figure 5.2: Cutting (left) and chopping (right).

While the parameters of this new skill can be refined by further training, even allowing them to diverge from the grasping-inside parameters, the important point is that the new skill is learned much more efficiently based on a prior skill than if it were learned from scratch.

If the classification functions do not diverge, we now have one category that applies to distinct skills. We may make this explicit by relabeling the category from *graspable-inside* to the more general *hollow*.

Next, the agent may observe fish being cut with a knife (Fig. 5.2, left). None of this resembles anything the agent has seen before, so it has to learn new classification and regression functions as described above, aided by its own exploration which provides negative and additional positive training examples.

Note that this example goes beyond the above in that two objects are involved, allowing the robot to learn about cutting tools as well as about cuttable objects.

Then, imagine the agent observes leek being chopped with a cleaver (Fig. 5.2, right). Similarly to the transfer from grasping-inside to dropping-into, the agent can observe, by statistical analysis of observed features and relations, that both cutting and chopping both involve flat object parts (*blades*) positioned appropriately with respect to the object being sliced. This allows a generalization to all unseen objects that exhibit features correlated with a blade.

Moreover, the agent can observe that cutting and chopping share action parameters (a transversal, up-and-down blade movement with respect to the object being sliced), while other parameters are distinct

(cutting involves a longitudinal movement which chopping does not). Similarly to the transfer from grasping-inside to dropping-into, this allows the robot to derive chopping actions from cutting actions, and vice versa. This allows the agent to use any unseen object with a blade for cutting and for chopping. Again, these bootstrapped skills may diverge under further training, leading e.g. to the realization that certain objects are better chopped than cut, and so on.

5.2 Learning object-action relations

Learning object and action relations is a difficult task. The difficulty comes from two main sources. First, the structure of descriptions of particular objects is very complex, while those descriptions are generally incomplete. The descriptions are derived from several sources, and the corresponding feature spaces are high-, even infinite-dimensional. Some features possess intriguing internal structure, e.g. graphs, which require computationally intensive preprocessing. The second source of difficulties is the small number of experiments which can confirm our hypotheses about the relationships between objects and actions. The experiments might even provide contradicting outcomes; thus the reliability of our knowledge is limited. We outline a learning approach to overcome on these difficulties by synthesizing some known methods and extending them by new elements to form an efficient learning framework. More detail is given in [Sze12].

Another approach is to make use of elaborate structures of the sensory space, in particular vision. This concerns in a separate modeling of visual modalities and a deep hierarchical representation of visual information providing an efficient feature space for learning. In [14], we have presented relevant knowledge about the structure of representation of visual information in primates for computer vision scientists. The ECV system described in [13, 17] and [MPK13] provides visual information in a structured way similar to the primate's visual system. As it is made explicit in the attachment [MPK13], this structure can be utilized to apply learning methods that identify individual features or feature dimensions relevant for certain tasks. These methods can be related to feature selection (such as the the Random Forest approach in appendix [MPK13]) or to the a prior algorithm [1].

To solve the learning problem drafted above we need to synthesize methods which fully or partially address the complexity of the data representation and the same time incompleteness and the sparsity of the available data sources. Here some possible approaches are summarized.

A general framework to learn sparse incomplete relations between several data sources is introduced in [20]. That general framework has been applied for recommender systems in [10] and [9]. The recommender systems connect users to objects, e.g. books, movies etc., and have to tackle problems of very high level of sparsity, since most of the user-object pairs are missing. Very frequently less than one percent of pairs can be observed; therefore the other ones need to be predicted. We face similar problem in learning the relationships between actions and objects, only a very small number of actions and objects can be tried in real experiments. Therefore, predicting the outcome of an experiment executed on an action-object pair can borrow the approach applied for the recommender systems.

Looking the problem from another angle we can formulate the learning task as an instance of so-called *transfer learning*; see a survey of approaches for example in [16]. In transfer learning, similar learning tasks are connected by assuming that if a method can be applied on one problem **P** then the accuracy of the prediction can diminish only a little bit if the problems have similar structure. The structural similarity means that the underlying unknown distributions of those problems are sufficiently similar. By applying transfer learning, the amount of experimental data can be reduced, or in other words a prediction derived on a subset of experiments can be propagated towards cases of action-object pairs having similar properties. One of the subbranches of the transfer learning is called *multitask learning*; see details in [7], where kernel methods are used to connect different learning tasks.

One can address the sparseness by applying methods of semi-supervised learning; see a survey in [27] and a collection of papers in the book [6]. In semi-supervised learning, it is assumed that a large sample of unlabeled cases is available. The internal structure of the space of those cases can then be exploited to propagate the knowledge about the labeled, annotated, cases in the entire database. A concrete example: consider the case where a large number of objects is available, but the number of actions, and the tried action-object pairs, is small. If the structure of the space spanned by the objects is sufficiently rich, then the action-specific properties of the objects can be transformed into each other whenever those objects are sufficiently similar. Then, carrying out an action on one object can imply the potential outcome of that

action on similar objects. In this way the action can be propagated among the objects. See some details about label propagation applied in semi-supervised learning in [5]. An example where the manifold type structure of the unlabeled cases is exploited can be found in [3] and [19].

We also need to address the complexity of the data representation. Here the objective is to find a simple model which approximates the original representation sufficiently well. It falls close to the variational Bayesian methods, where complex distributions are approximated by simple ones. Particular techniques can extract a subset of observed features, or derive new, higher level features out of the elementary observations, see methods of so called deep learning described for example in [4].

5.3 General concept of an implementable learning system

The central building blocks of the learning system, the databases and the algorithms processing those databases to provide prediction associating object and actions, consist of the following main elements.

The learning system is built upon two databases. One contains the objects that can be used as tools or targets in an action; the other database collects the possible actions. Both the objects and the actions are characterized by collections of features. Those collections will be called *object* and *action profiles* in the sequel. The databases are built upon these profiles. An example of a database element can be seen in Figure 5.3.

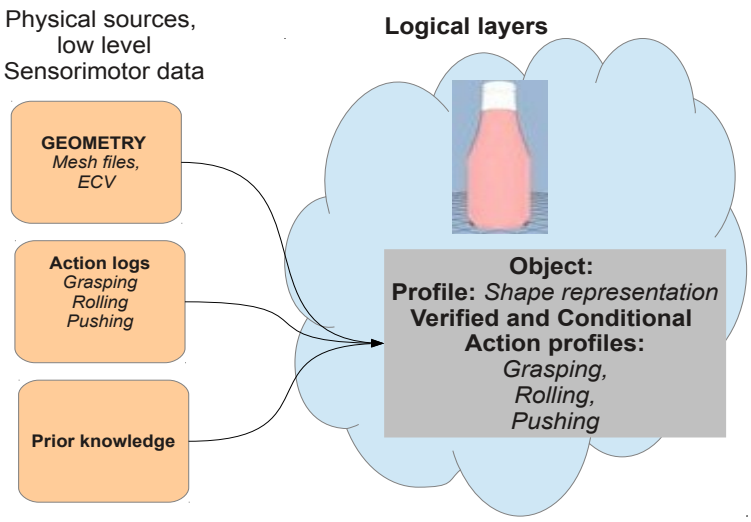


Figure 5.3: An element of a layer of the object database

Based on the features of the objects and actions we can construct a space of objects and similarly a space of actions. These spaces are endowed with an internal structure, a certain geometry, which allows us to compare the elements the objects and the actions. This geometry expresses the similarity, or dissimilarity of the elements. The similarity is derived from the corresponding features, and it might be realized by a conditional probability or a kernel function, but any similarity measure which admits efficient computation an sufficient predictive power can be considered.

The databases are layered; the database of objects has as many layers as actions are available. All layers can comprise different profiles to each object. Therefore the similarity measure can vary between layers as well, where every action implies its own space and geometry of the objects. An example of the action-related layers of object database is shown in Figure 5.4.

The geometry of those spaces, the similarity measure, allows the learning system to propagate the knowledge collected in the earlier experiments, namely: if an action $\mathscr{A}$ can be carried out on object $\mathcal{X}$ then with a certain probability that action could be applied on similar objects in the neighborhood of $\mathcal{X}$.

The object space provides the *exploration space* for the learning procedure; hence it allows new candidates to be selected for the new experiment in a controlled way. This approach can efficiently enhance statistical

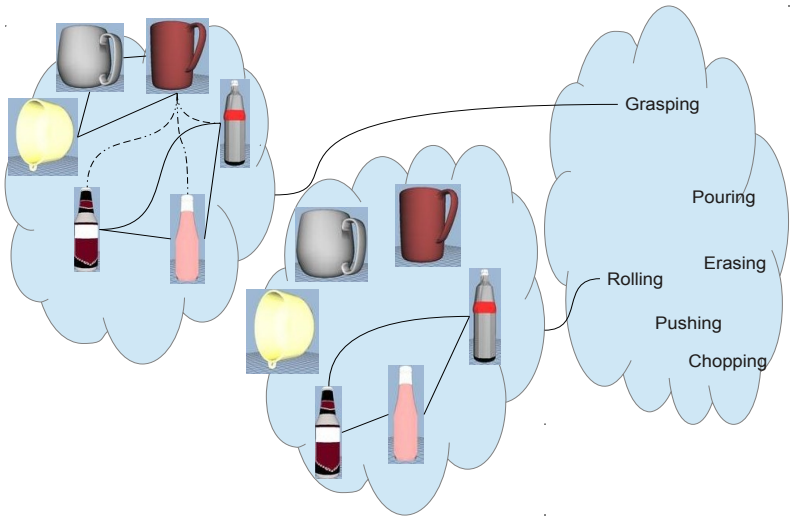


Figure 5.4: Action related layers of the object database. The lines in within the layers expresses the similarities between the objects.

stability of the learning procedure by eliminating entirely unknown objects whose behavior in any action could be unpredictable. On the other hand based on the space structure we can apply an active learning policy to design the new experiments.

The core of the learning model realizing the outlined system of action and object interactions has been implemented in Matlab and in Python as well. We are going to extend the latter one to guarantee the free dissemination of the model. The theoretical background of the implemented model is detailed in the Appendix of [Sze12].

5.4 Proof-of-concept implementation

A first implementation of core aspects of the system outlined above was implemented at UIBK based on maximum-margin regression. This proof of concept focuses on the scenario of a robot having to recognize objects that are the potential targets of its activities. This can be considered as a simple classification problem, but capturing the functional category, e.g. whether an object on the table can be used to cut something, is more important than the simple decision whether it is a knife. One of the most promising approaches is one that can combine the simple classification with the general functional categories. Recognizing a concrete object obviously can help to find its functionality, but when the number of distinct objects is large, e.g. > 100, then the accuracy of the object recognition can severely degrade. Therefore observing wider functional categories of the objects as additional information should improve recognition performance for robotic action.

Thus the learning task is to find the class (or category) and/or higher-level categories of an object, given shape descriptions of the objects. The categories are given as object labels, and they are organized into a hierarchy mainly based on the functional similarity of the objects. The functional similarity connects objects used for the same purpose, e.g. airplanes or items of furniture. This kind of functional similarities might also connect objects with significantly different shapes. The categories form a tree structure, where the base classes correspond to the leaves of that tree. In this structured representation the objects are formally annotated by a path connecting the root node of the category tree to the node corresponding to the given class label of that object. Such models are generally known in the literature as *structured learning*, see references in [Sze13].

To solve the learning problem outlined above we created an output kernel function, a graph kernel, expressing the similarities between the paths annotating the objects, and an input kernel function expressing the similarities between the shape of the 3D objects. The output kernel is built on the weighted node-wise similarities of the paths, where the weights can force higher similarities between the lower

level classes expressing the concrete specificity of the objects. The input kernel is computed on the local distribution of the point cloud representing the objects. These local distributions are then clustered to derive characteristic classes of the surface points of the shapes. This characterization can provide the basis for shape segmentation as well.

The optimization model implementing the learning problem is a version of Maximum Margin Regression (MMR). This model can find the functional dependency between the complex structured outputs, the paths of the category graph, and the inputs given by the shape features. MMR is a generalization of the well-known Support Vector Machine that can handle arbitrary outputs representable by kernels. The base version of the MMR model can be computed at the same level of computational complexity as the SVM, independently of the complexity of the output representation. An extended version of MMR is used to realize the action-object interactions.

Using these MMR methods, we produced indicative results on a small shape recognition problem involving kitchen objects, demonstrating that knives and cleavers can be recognized as the same or as distinct categories, as might be required by an evolving robot experience. A second, large-scale, hierarchical shape classification experiment confirms that structured prediction can in fact exploit category hierarchies to obtain functional predictions superior to base-category predictions. Although no directly comparable results exist in the literature, the available data testify to the highly-competitve nature of the proposed method [Sze13].

Further details concerning methods for implementing the general learning system of Section 5.3 are given in another attached technical report [Sze12].

5.5 Learning grasp and tool-use competences in the cross space of Early Cognitive Vision and Action Representation

In primates, vision plays a dominant role. Half of the primate's cortex is connected to visual tasks (see [14] for a review on the visual cortex for computer vision scientists), and it can provide high precision 3D shape information. In addition, vision provides rich appearance information in terms of colour and texture, all aspects which are of significant relevance for affordance computation.

The human visual system constitutes a deep hierarchy covering a large number of complementary aspects at different levels of granularity (see figure 5.5). Moreover, more than 2/3 of the visual cortex in the is associated to task-independent feature processing in the occipital cortex displayed as yellow areas in figure 5.5. In these areas, feature descriptors covering different aspects of visual information such as color, 2D and 3D shape as well as motion are extracted in – at least at early stages of processing – largely separated processing streams. In the hierarchical process, the level of abstraction of feature representation as well as the receptive field size increases. For example in the colour domain, on the retinal level a trichromatic colour representation with a receptive field size of 0.01 degree[1] while on the level of V1, color is encoded in an opponent space with a receptive field size of 3 degree while on the level of V4, hue as an object property with high degree of color constancy is coded by neurons with a receptive field size of up to 8 degrees. Similar transformations also occur in other domains, for example zero–th, first and second order disparity (corresponding to absolute distances, tilt and curvature respectively) is coded at different levels of 3D shape processing. For details, we would like to refer the reader to [14].

In the last decade, SDU has made a substantial effort to derive a functional model of these early stages of the deep visual hierarchy covering the occipital areas of the primate's vision system. This has lead to the so called Early Cognitive Vision (ECV) system (see figure 5.6) presented in, e.g., [17, 13]. Structural bootstrapping in the sensorimotor domain can take advantage of relevant patterns in the ECV space that are highly correlated with action successes or perceptual categories.

As already pointed out by Granlund [11], the visual space is fundamentally of higher complexity compared to the action space. This in the first place concerns the dimensionality of visual information compared to a still low dimensionality of action parametrization connected to the rather limited number of joints to be actuated. As a consequence, affordance learning has to fundamentally deal with the high dimensionality of the space spanned by the visual hierarchy as discussed already in section 5.2.

In attached papers [TBK12, TK12], we present an algorithm which identifies relevant feature and action

[1] All measures are given for 5 degree of eccentricity.

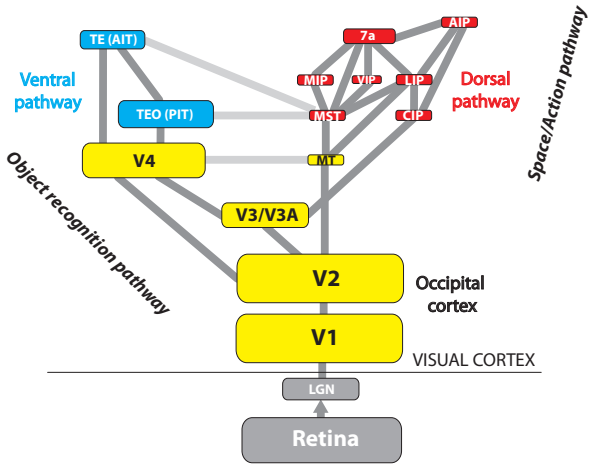


Figure 5.5: Hierarchy of the primate's visual cortex.

parameters in the context of affordance learning in the cross space of the space spanned by the ECV system and the action space (see also the examples in section 5.1). Up to now, we focus on grasping and cutting actions. Based on a set of evaluated actions, the algorithm identifies areas in the ECV x Action space which contain a large percentage of successful actions. A striking property of the algorithm is that it is able to identify both, the visual features that characterize the affordance *and* the action parameters associated to it, which is an important aspect in the concept discussed in section 5.3.

This algorithm is rather simple (being a variant of the a priori algorithm [1]): In the vicinity of a successful action, we search for (or generate) other successful action. In general terms, the algorithm resembles the 'search for a needle in the haystack' since only few of the dimensions of the ECV space – and in these dimensions only very small subspaces – are relevant.

We demonstrate the performance of the algorithm on the generation of grasp affordances triggered by visual feature relations (see figure 5.7) and are in the process of applying it also to the learning of a 'cutting affordance', both according to the examples described in section 5.1. In the case of 'grasping', we learn the visual features that trigger grasp options as well as how to perform the actual grasp action in relation to these features. In case of 'cutting', we learn the shape structure that defines a cutting tool as well as the way of performing the actual cutting action as described in [PSJ+12].

The complexities associated to the approach stems from four sources:

Generation of Sufficient Percentage of Successful Actions: Non-successes are mostly irrelevant — although these are used for sorting out non–interesting areas — and hence the system needs to be able to perform actions with a certain percentage of success likelihood. This can be achieved by introducing bias, e.g., by designing reflexes which trigger actions with sufficient success likelihood (as in, e.g., [13]) or by means of a 'Learning by Demonstration' setting.

Spanning a feature space which provides features that are indicative for actions: The most challenging problem is to provide a feature space which covers features that are highly correlated to successful actions. Which features are relevant might depend significantly on the actual task and most features will be actually highly uncorrelated to action successes and therefore insignificant. Actually, an important aspect is to separate important from non-important sub–areas of the ECV x Action space. The ECV space — besides covering different aspects of visual information and providing descriptors of different granularity — also provides feature relations of different order. This very quickly leads to a dimensionality which cannot be explored exhaustively (dimensionality explosion). It is also important to keep in mind that the relevant features are not necessarily to be found at high levels of the hierarchy (for example to distinguish a sawing blade from a cutting blade requires high frequency information processed at low levels of the hierarchy).

Nevertheless, we argue that the ECV space provides some important properties that facilitate the identification of relevant sub–areas. Most importantly, the separation of visual aspects allow for disregarding of irrelevant dimensions. Note that, while on the pixel level different visual modalities are still highly intermingled, it is an important task of visual processing to allow for this separation

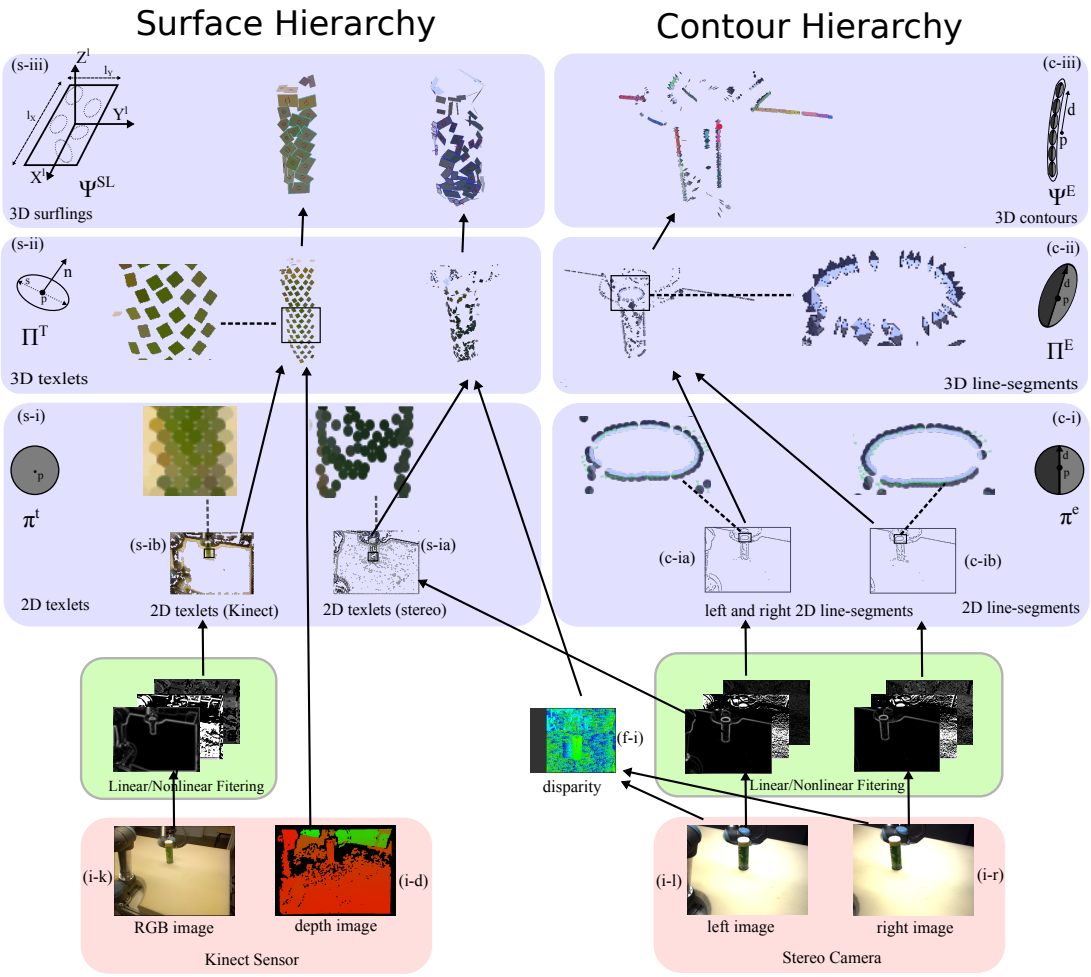


Figure 5.6: The hierarchical representation of edge and texture information in the ECV system. (i-l,i-r) shows an example stereo image pair while (i-k,i-d) are the RGB and the depth images from Kinect sensors. (c-i) 2D line segments for the left ((c-ia)) and the right ((c-ib)) image. (c-ii) 3D line segments. (c-iii) 3D contours. (s-i) 2D texlets for the left image ((s-is)) and from Kinect ((s-ib)). (f-1) disparity image from stereo images. (s-ii) 3D texlets. (s-iii) 3D surflings. This figure is best viewed in color.

already at rather early stages of the processing pipeline such that irrelevant dimensions can be easily disregarded. Another important aspect is the deep hierarchy which provides information on different levels of granularity. The learning can then pick the relevant level of granularity for a given task.

Defining a metric on the combined feature-action space: The algorithm searches for *similar* events in the *vicinity* of successful feature-action associations. Hence a metric needs to be defined on the ECV x action space. This metric might be different for different tasks. Also the metric can be used to disregard irrelevant dimensions (see above).

Large spaces need large data to be filled: Our algorithm — although (or maybe since) being simple — requires certain amounts of data to lead to statistical statements that are significant. This requires 'number crunching' as well as 'large storage capacities'. Both is — in a somewhat distressing way — provided by the human visual system with the large resources devoted to early cognitive processing in the occipital cortex. Maybe one of the reasons of the success of human cognition is not in the 'high complexity of algorithms' but the ability to deal with such 'big data'. Note that also in other fields than vision such as, e.g., speech recognition, success stories could be drawn not by increasing the complexities of models but in the contrary, by the the simplifications of models and the use of huge amounts of data. The reason for that might be that with the sophistication of the models potentially sub-optimal bias enters the process.

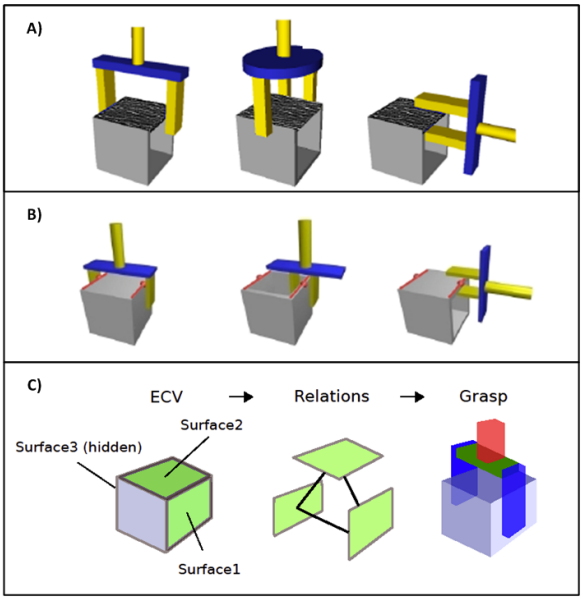


Figure 5.7: Learning grasp affordances.

In [TBK12], we describe the abstract learning approach and in [TK12], we give results on grasp learning. In [MPK13], we describe a search algorithm for visual features only (in the ECV space) for object recognition.

5.6 Progress in implementations, integration and results

A detailed description of the learning system as outlined in Section 5.3 is given in [Sze12]. Its appendix contains a detailed description of learning algorithms that implement core aspects of this learning system.

The consortium is in the process of drafting an experimental agenda to explore and demonstrate bootstrapping at sensorimotor and object-action levels [PSJ+12]. That document briefly summarizes the targeted structural-bootstrapping functionalities, and outlines a sequence of increasingly ambitious experiments. The first of these experiments addresses categorizing objects by affordances. For this first experiment, affordances are provided by manual annotations; they are not grounded in actions yet. Thus, the learning problem is reduced to a shape-based object categorization problem. Experimental results are given in an attached technical report [Sze13].

The next step involves adding kinematic simulation of cutting actions, using these to derive implicit functional categorizations of objects, and demonstrating the transfer of action parameters between objects. Suitable kinematic simulation facilities have already been developed and implemented at SDU [PSJ+12].

Future work will ultimately add physical action, and will address more complex aspects of structural bootstrapping, as described in the experimental agenda [PSJ+12]. This agenda is intended to be an evolving document; the version included represents a current snapshot.

Methods and implementations for various subproblems of the integrating learning system described in this chapter are under development by various partners within the consortium:

- Inter-object action transfer and the adaptation of known actions to other, similar actions is addressed by the sensorimotor structural bootstrapping method combining statistical and reinforcement learning reported in Chapter 3.
- Refinement of action sequences on the basis of known action sequences and instantiation of new action sequences, described at the level of parametrized relations between objects, is addressed by the SEC assimilation and accommodation method reported in Chapter 4.
- The system outlined in this chapter operates on in-memory object models. This is a key property

of this system, as it alleviates the need for high-performance sensing able to provide high-fidelity reconstructions of the robot's environment. Following this idea and similarly to human routine behavior, the agent possesses fairly detailed internal models of known objects on which most reasoning is performed, while perception merely links the real world to the internal models. Several suitable perceptual systems are available within the consortium. New methods linking 2D observations to 3D models are under development at UIBK [21, 22, 23, 24], as well as for robust and efficient 3D shape registration [25]. These will be mentioned in the upcoming Year-2 Periodic Progress Report, and full detail will be given in Deliverable D2.1.2 (Month 37).

- In an open world, objects need to be added to the database dynamically. This can be done via dedicated exploration, under more control over the explored object than during casual interaction, involving picking up the object, looking at it from several, close-up viewpoints, etc., just as humans would do. KIT has recently developed methods suitable for the discovery and exploration of unknown objects [18], which will likewise be reported in the upcoming Progress Report.

- Object similarity functions tunable toward specific actions are a crucial ingredient of the system described in this chapter. Anticipating that action-specific object similarity functions will benefit from explicitly-identifiable object parts, UIBK is working on automatic discovery of common object parts within a structural latent-variable model [26]. Again, this work will be reported in the upcoming Progress Report and in D2.1.2.

- SDU has developed an algorithm which searches in the ECV×Action space for correlated shape-action events. Promising experimental results for the case of grasping were achieved; current work focuses on cutting. SDU also approached the problem of finding and utilizing features for object recognition in the ECV space only. This work is described in attached papers [TBK12, TK12, MPK13].

Chapter 6

Conclusions

We have defined a promising, overarching framework for object-action structural bootstrapping (Ch. 5) that postulates concrete computational mechanisms for the core aspects of structural bootstrapping – leveraging past experience to boost learning on future tasks by discovering similarities between objects and between actions, and making inferences about novel objects and actions based on past experience. It will incorporate sensing and action from other work packages, as well as sensorimotor and event-chain learning, spanning multiple levels of abstraction and including important contributions from multiple project partners.

Some, but not all of the details of the framework have already been fleshed out in detail [Sze12]. We have created a roadmap for experiments to guide future development [PSJ+12], and have produced results for the first step of this roadmap [Sze13]. Future work will follow this roadmap. The next step involves simulated interaction to replace the category labels of our initial results, and augmenting the scope of the learning task to include action parameters.

References

[1] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[2] E. E. Aksoy, A. Abramov, J. Dörr, K. Ning, B. Dellen, and F. Wörgötter. Learning the semantics of object-action relations by observation. *The International Journal of Robotics Research*, 30(10):1229–1249, 2011.

[3] M. Belkin, P. Niyogi, and V. Sindhwani. On manifold regularization. In *AISTATS*. 2005.

[4] Yoshua Bengio. *Learning Deep Architectures for AI*. NOW Publisher, Foundations and Trends in Machine Learning, 2009.

[5] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. Label propagation and quadratic criterion. In *Semi-Supervised Learning*, pages 193–216. MIT Press, 2006.

[6] O. Chapelle, B. Schölkopf, and A. Zien Editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2010.

[7] T. Evgeniou, C.A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research, 6(Apr)*, pages 615–637, 2005.

[8] Severin Fichtl. Making artificial intelligence which copies the way babies learn. Master's thesis, University of Aberdeen, 2011.

[9] M.A. Ghazanfar, A. Prugel-Bennett, and S. Szedmak. Kernel mapping recommender system algorithms. *Information Sciences*, 2012. accepted.

[10] M.A. Ghazanfar, S. Szedmak, and A. Prugel-Bennett. Incremental kernel mapping algorithms for scalable recommender systems. In *IEEE International Conference on Tools with Artificial Intelligence (ICTAI), Special Session on Recommender Systems in e-Commerce (RSEC)*. 2011.

[11] G.H. Granlund. The complexity of vision. *Signal Processing*, 74, 1999.

[12] F. Guerin, N. Krüger, and Kraft D. A survey of the ontogeny of tool use: from sensorimotor experience to planning. *Autonomous Mental Development, IEEE Transactions on*. (accepted).

[13] G. Kootstra, M. Popovic, J. Jørgensen, K. Kuklinski, K. Miatliuk, D. Kragic, and N. Kruger. Enabling grasping of unknown objects through a synergistic use of edge and surface information. *The International Journal of Robotics Research*, 31(10):1190–1213, 2012.

[14] Norbert Krüger, Peter Janssen, Sinan Kalkan, Markus Lappe, Aleš Leonardis, Justus Piater, , Antonio J. Rodríguez-Sánchez, and Laurenz Wiskott. Deep hierarchies in the primate visual cortex: What can we learn for computer vision? *IEEE PAMI*, accepted.

[15] T. Kulvicius, K. J. Ning, M. Tamosiunaite, and F. Wörgötter. Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on handwriting. *IEEE Transactions on Robotics*, 28(1):145–157, 2011.

[16] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22/10:1345–1359, 2010.

[17] N. Pugeault, F. Wörgötter, and N. Krüger. Visual primitives: Local, condensed, and semantically rich visual descriptors and their applications in robotics. *International Journal of Humanoid Robotics (Special Issue on Cognitive Humanoid Vision)*, 7(3):379–405, 2010.

[18] David Schiebener, Julian Schill, and Tamim Asfour. Discovery, segmentation and reactive grasping of unknown objects. In *IEEE-RAS International Conference on Humanoid Robots*, 2012.

[19] V. Sindhwani, M. Belkin, and P. Niyogi. The geometric basis of semi-supervised learning. In O. Chapelle, B. Schlkopf, and A. Zien, editors, *Semi-supervised Learning*, pages 217–235. MIT Press, 2006.

[20] S. Szedmak, Y. Ni, and S. R. Gunn. Maximum margin learning with incomplete data: Learning networks instead of tables. *Journal of Machine Learning Research, Proceedings*, 11, Workshop on Applications of Pattern Analysis:96–102, 2010. jmlr.csail.mit.edu/proceedings/papers/v11/szedmak10a/szedmak10a.pdf.

[21] Damien Teney and Justus Piater. Generalized Exemplar-Based Full Pose Estimation from 2D Images without Correspondences. In *Digital Image Computing: Techniques and Applications*, 2012.

[22] Damien Teney and Justus Piater. Sampling-based Multiview Reconstruction without Correspondences for 3D Edges. In *3DimPVT*, pages 160–167, 2012.

[23] Damien Teney and Justus Piater. Continuous pose estimation in 2D images at instance and category levels. 2013. Submitted.

[24] Damien Teney and Justus Piater. Modeling Pose/Appearance Relations for Improved Object Localization and Pose Estimation in 2D images. In *6th Iberian Conference on Pattern Recognition and Image Analysis*, LNCS, Berlin, Heidelberg, New York, 6 2013. Springer. To appear.

[25] Hanchen Xiong, Sandor Szedmark, and Justus Piater. Efficient, general point cloud registration with kernel feature maps. 2013. Submitted.

[26] Hanchen Xiong, Sandor Szedmark, and Justus Piater. Learning 3D part-based models of object categories with robust alignment and consistent segmentation. 2013. Submitted.

[27] X. Zhu. Semi-supervised learning literature survey. *Computer Science Department, University of Wisconsin, Madison*, 2006. www.cs.wisc.edu/ ∼jerryzhu/pub/ssl_survey.pdf.

Attached Articles

[ATV+13] E. E. Aksoy, M. Tamosiunaite, R. Vuga, A. Ude, C. Geib, M. Steedman, and F. Wörgötter. Structural bootstrapping at the sensorimotor level for the fast acquisition of action knowledge for cognitive robots. Technical report, University of Goettingen, 2013.

[FAG+12] S. Fichtl, J. Alexander, F. Guerin, J. A. Jørgensen, D. Kraft, and N. Krüger. Rapidly learning preconditions for means-end behavior using active learning. In *IEEE Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, 2012.

[FAK+] S. Fichtl, J. Alexander, D. Kraft, J. A. Jørgensen, N. Krüger, and F. Guerin. Learning object relationships which determine the outcome of actions. *Paladyn*. (submitted).

[MPK13] Wail Mustafa, Nicolas Pugeault, and Norbert Krüger. Multi-view object recognition using view-point invariant shape relations and appearance information. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[NFV+12] B. Nemec, D. Forte, R. Vuga, M. Tamosiunaite, F. Wörgötter, and A. Ude. Applying statistical generalization to determine search direction for reinforcement learning of movement primitives. In *12th IEEE-RAS International Conference on Humanoid Robots*, pages 65–70, Osaka, Japan, 2012.

[PSJ+12] Justus Piater, Sandor Szedmak, Jimmy Jørgensen, Dirk Kraft, and Norbert Krüger. Object-action structural bootstrapping. Technical report, University of Innsbruck, 2012.

[Sze12] Sandor Szedmak. Learning object-action relations via knowledge propagation. Technical report, University of Innsbruck, 2012.

[Sze13] Sandor Szedmak. Shape-based object categorization. Technical report, University of Innsbruck, 2013.

[TBK12] M.T. Thomsen, L. Bodenhagen, and N. Krüger. Statistical identification of composed visual features indicating high likelihood of grasp success. *SDU Report*, 2012.

[TK12] M.T. Thomsen and N. Krüger. Finding the needle in the haystack: Identifying relevant feature–action associations. *SDU Report*, 2012.