Involving behavior in the formation of sensory representations

Dissertation

zur Erlangung des Grades

“Doktor der Kognitionswissenschaft“

(„PhD in Cognitive Science“)

Im Fachbereich Humanwissenschaften

der Universität Osnabrück

Vorgelegt von

Daniel Weiller

Eisenbahnstrasse 14
49074 Osnabrück
Geb. 8.01.1977 in Landau/Pfalz

Osnabrück, February 2009

Attributions

In this section a list of projects and project participants are given that contributed to this thesis. The communication between the Khepera robot and the MicorPsi Program (sections 2.1.5.3 and 2.1.5.4) was outlined by Leonhard Läer and me, implemented by Leonhard Läer (see Master thesis of Leonhard Läer: ”Intelligent foraging of an agent acting in the real world”), while the tracking system (2.1.5.2) was implemented by me. The agent architecture (2.1.1) was outlined and discussed within the Navigation study project, participated in by Cliodhna Quigley, Christian Mühl, Isabel Dombrowe and me. The implementation of this agent in the MicroPsi framework (2.1.5.6) was done by Cliodhna Quigley, Christian Mühl and me. Further, the modification of the agent to the four-arm-maze task (section 2.1.6), batch and online learning procedure (2.1.3), the recording and the analysis of the robot behavior (section 2.2) was done by me. The work presented in section 3 is based on the Master thesis of Robert Märtin and Bachelor thesis of Sven Dähne (“Unsupervised of optimizing sensorimotor space representations”). The data used in section 3 is based on a modification of the reorganization algorithm which was implemented by me. Further, Priska Herger successfully applied the developed optimization scheme to the transition probabilities generated by a real robot interacting with its four-arm-maze environment in her bachelor thesis (“Optimizing sensorimotor state space of a physical robot model”). This data is not integrated in this thesis. Also this thesis does not contain any data recorded during a collaborative project with Joerg Hipp and Christiane Vahle-Hinz, in which local field potentials from the barrel cortex were recorded while the corresponding whisker was stimulated with natural stimuli.

List of supervised bachelor and master thesis

Leonhard Läer: “Intelligent foraging of an agent acting in the real world.”
Robert Märtin: “Unsupervised optimization of sensorimotor space representations”
Sven Dähne: “Unsupervised optimizing of sensorimotor space representations”
Priska Herger: “Optimizing sensorimotor state space of a physical robot model”

List of publications

Weiller, D.; Läer, L.; Engel, A.K.; König, P.: Unsupervised learning of reflexive and action-based affordances to model navigational behavior. Proceedings KogWis2007, Saarbrücken (2007)

Weiller, D.; Läer, L.; Engel, A.K.; König, P.: Unsupervised learning of reflexive and action-based affordances to model adaptive navigational behavior. Submitted to Autonomous Robots.

Weiller, D.; Märtin, R.; Sven, D.; König, P.: Involving motor capabilities in the formation of sensory space representations. In preparation

The manuscripts of the first two publications can be found in the Appendix.

Introductory remarks

Parts of this thesis (text and figures) were published, submitted or are in preparation for publication in different scientific journals. Most of the material presented in this thesis is part of these publications. The different chapters of this thesis correspond to the publications listed below.

Reference	Chapter
Weiller et al., 2007; Weiller et al., (submitted);	2
Weiller et al., (in preparation)	3

Abstract

In computational neuroscience, the relation between neural response properties and the statistical properties of the neuron's sensory input has recently moved into the centre of interest. Neurons are sensitive to specific aspects of natural stimuli, which are – according to different statistical criteria – an optimal representation of the natural sensory input. Since these statistical criteria have only been applied to an agent's sensory input, it is still an open question whether these purely sensory representations are suited to generate meaningful behavior. In the present work we introduce an optimization scheme that applies a statistical criterion to an agent's sensory input while taking its motor behavior into account. We first introduce a general cognitive model that spans a variety of behavioral domains, and second develop an optimization scheme that increases the predictability of the sensory outcome resulting from an agent's motor actions. Finally, we outline the application of the optimization algorithm and cognitive model within a navigational paradigm.

In the cognitive model, place cells served as a suitable low-dimensional sensory representation, dividing the environment into discrete states similar to the place fields of hippocampal place cells. The agent randomly explored its environment and learned the structure of its habitat by memorizing the sensory outcome of its actions. This learning was composed of a central process, learning state-to-state, i.e. place-to-place, transition probabilities in an environment, and a distal routine, learning the extent to which these motor actions are caused by sensory-driven reflexive behavior (obstacle avoidance) rather than by central processing. Navigational decision making integrates both centrally and distally learned sensory knowledge from the environment to derive the actions that are most likely to lead to a navigational goal. This cognitive scheme was implemented on a robot navigating in a four-arm maze. Furthermore, as a next step we introduced an optimization process that modified the state distributions to increase the predictability of the sensory outcome of the agent's actions. This heuristic, rule-based optimization algorithm was tested with respect to changes in the environment as well as to changes in the motor capabilities of the agent.

The cognitive model successfully performs the navigational task, and the differentiation between central and distal processing increases both general behavioral accuracy, as well as behavioral adaptation to changes in the environment. This cognitive model can be easily modified to model a variety of behaviors, like the standard four-arm-maze task demonstrated in this work. Further, the optimization of the sensory states resulted in spatially compact and round states, similar to place fields found in behaving animals. The spatial distribution of states depends on the agent's motor capabilities as well as on the environment. We compared place fields resulting from optimizing the statistical criteria of temporal coherence, which take only sensory input into account (Wyss et al., 2006) to those resulting from our approach, which takes sensory and motor input into account. The place fields from our approach showed both a high predictability and a temporal coherence similar to those of Wyss and colleagues, but in contrast the latter had a low predictability. Thus, our approach combines the advantage of temporal coherence with behavioral relevance, and furthermore the general coding principle of predictability entails temporal coherence. Our results suggest that the agent's motor apparatus can play a profound role in the formation of place fields that are optimal for an actively behaving agent. Hence, sensory representation in higher cortical hierarchies should be understood by combining sensory input with the motor capabilities of a behaving agent.

1. Introduction

In recent years, the field of sensory coding has successfully demonstrated how the brain adapts to its environment through development and learning. A substantial number of non-trivial properties of sensory neurons can be understood by applying general coding principles to the specific regularities found in the natural sensory input of various organisms (Simoncelli and Olshausen, 2001; Körding et al., 2004; Hopfield, 1982; Willshaw and von der Malsburg, 1976). Sensory coding principles (Barlow, 1961) pose the reduction of redundancy in the sensory input as a normative objective for sensory neurons, and allow for the description of invariance in neuronal responses to sensory input. Applying the most prominent coding principles – sparse or efficient coding and temporal stability, as well as the recently developed predictability – to passive sensory input from different modalities, yielded important insights into the organization of an organism's sensory space. Even more recent theoretical developments additionally emphasize the involvement of an organism's action in the organization of its sensory space (König and Krüger, 2006; Philipona et al., 2004; O'Regan and Noe, 1996; Hommel et al., 2001; Grush et al., 2004), and as a corollary, also the role of actions in determining response properties of sensory neurons. These theories highlight the importance of relating neuronal response properties not only to the sensory space, meaning the space of an organism's possible sensory input, but rather to the sensorimotor space, which is the space of an organism's possible sensory input and possible motor commands. In neuroscience, such a shift in perspective has its foundation in recent findings regarding the involvement of cortical motor areas in perception and sensory processing. Here we try to use the coding principle of predictability in order to reorganize an agent's sensorimotor space in the context of its well-defined behaviors.

In the next section we will first examine the foundations of sensory coding. As we seek to apply normative coding principles not only to sensory space but to sensorimotor space, we then introduce the concept of sensorimotor space as has been applied in the neurosciences, followed by an introduction of computational concepts used to specify behavior in sensorimotor space. The remainder describes the main aim of this thesis: to use normative coding principles to optimally reorganize the sensorimotor space of a navigating agent by increasing its ability to predict the sensory outcome of its own actions.

1.1 Relation of sensory input and neural response

Theories of sensory coding aim to relate properties of sensory neurons to their sensory input, thus treating neuronal processing as a mechanism to optimally encode sensory signals. We will start with some evidence for the importance of the role of neuronal activity in the development of neural circuits. Spontaneous fluctuations of the electrical signal and experience-driven, stimulus-related activity are both important for brain development. Part of experience-driven neuronal activity can be understood from the statistical regularities inherent in the sensory input, i.e. from regularities contained in the stimuli naturally encountered by an organism, which compose a special subset of all possible sensory stimuli. Hence we describe the statistical properties of natural stimuli, specifically in the visual system. The end of this section then describes how sensory coding principles can explain different properties of sensory neurons based on sensory input.

1.1.1 Sensory input and neural development

The contribution of spontaneous and experience-driven activities to the development of early sensory systems has been the subject of many investigations. For example, on the one hand, spontaneous activity plays a crucial role in the development of orientation selectivity (Penn et al., 1999; Wiesel and Hubel, 1974) in primary visual cortex (V1), as shown in the seminal work of Hubel and Wiesel (Hubel and Wiesel, 1968). On the other hand, experience-driven activity, generated by visual sensory input, refines and matures the orientation-selective properties in response to sensory stimulation (Chapman et al., 1993). In addition, Imbert and colleagues (Imbert et al., 1975) demonstrated that the prolonged deprivation of visual input results in a progressive deterioration of the receptive fields properties in V1. Such reorganization of sensory cortices by neural activity has also been shown in non-visual modalities, for instance the primary tactile cortices which receive input from the rat whisker system. Removal of whiskers prevents the formation of the barrel structure that is typical in the rodent primary tactile cortex (van der Loos, 1973). Thus spontaneous as well as experience-driven activities are necessary for the normal development of sensory cortices.

Further studies aimed to demonstrate that the modality of a sensory cortex is essentially determined by experience-driven processes, and not pre-specified by anatomical connections. In so called “rewiring” experiments (Sur et al., 1988; for review Sur and Leamey, 2001), visual input normally targeting the primary visual cortex was rerouted to the auditory cortex, and was found to lead to receptive field properties in the rewired auditory cortex that are in many ways similar to cells in primary visual cortex. Although the “auditory visual cells” showed larger receptive fields (Roe et al., 1992), they possessed typical visual properties like orientation, direction and velocity selectivity as well as a spatial organization typical for the primary visual cortex. Even more convincingly, behavioral experiments could also show that the neurons in the rewired auditory cortex are indeed functionally relevant for visual driven behavior (von Melchner et al., 2000). These experiments demonstrate that auditory and visual primary cortices share innate similarities, which develop into the concrete neural response properties of a modality due to the influence of the statistical regularities of the respective sensory input.

1.1.2 Natural stimuli as sensory input

What are the statistical properties of sensory input that drive the development of sensory response properties? Currently there is a large body of literature investigating the relation of visual receptive field properties to the statistics of natural visual stimuli. Theoretically, there is an infinite amount of possible visual stimuli, which could be generated by assigning all possible combinations of numeric values to the pixels of an image. Yet, evolved in a natural environment the visual system only encounters a subset of all possible visual stimuli – the natural stimuli perceived in everyday life.

Hence, although the use of simple stimuli like oriented moving gratings has yielded important insights into the visual system, a growing community has recently begun to seek to understand structural similarities in the subset of natural stimuli in order to investigate the properties of the early visual system (for a review see Kayser et al., 2004). For example, Smyth and colleagues (Smyth et al., 2003) found that natural stimuli facilitate contextual

effects and non-linearities in the neuronal response which cannot be found or studied in the presence of simple stimuli. Such "non-classical" receptive fields have a great influence on sensory processing in the primary visual cortex, as shown by Vinje and Gallant (Vinje and Gallent, 2000, 2002).

The most prominent statistical property of natural scenes is a fall in their power spectrum with a slope of around $1/f^2$, with f as the spatial frequency, (Tollhurst, 1992; Ruderman and Bialek, 1994) believed to result from scale invariance in the visual world. That is, the statistical properties of images do not change with the scale of the observation. Further, compared to randomly-generated visual stimuli, these natural scenes are highly structured and neighboring pixels show a high correlation (Simoncelli and Olshausen, 2001). Apart from second-order statistical properties, natural visual input also has higher-order moments. This becomes obvious when removing second-order statistics by applying a whitening procedure to natural scenes. After this procedure the images still show a lot of structure which can be perceived by humans, such as edges. In addition, Dong and Atick (Dong and Atick, 1995) showed an interdependence between the spatial and temporal properties of natural visual input, related to movements of objects and of the agent in the world. In general, an analysis of the space-temporal characteristics is far more complex than the spatial alone, because obtaining time-varying retinal images requires the tracking of the eye, head and body of an agent interacting with the environment. Although this spatiotemporal aspect has not yet been fully characterized, it is true to say that natural visual stimuli have highly prominent statistical characteristics.

1.1.3 General coding principles applied to natural stimuli

As described above, the statistical properties of the sensory input given by natural stimuli are relevant aspects for the development of sensory cortices. In particular, generic coding principles have been found useful to explain neuronal response properties when applied to natural stimuli.

1.1.3.1 Sparse coding

In order to process the visual sensory input, a large number of neurons are required to represent each visual pattern contained in visual natural stimuli. Based on Shannon's information theory (Shannon, 1948), Barlow (Barlow, 1961) constrained the neural processing to be information efficient, also known as efficient coding.

In order to maximize the information transmitted by an individual neuron, without any constraints to its firing rate, neural responses over time should be uniformly distributed between the maximally and minimally possible neural responses. That is, neuronal information should be coded such that all responses occur equally often, allowing the neuronal channel to code a maximum of information. Such a uniform distribution was first found in the contrast response of neurons in the fly's visual system (Laughlin, 1981). Further, information theory predicts an exponential distribution of neural responses at a given fixed-mean firing rate, as found in the instantaneous firing rates of neurons in primary and temporal inferior cortices of cats and monkeys (Baddeley, 1997). These pieces of physiological evidence support the validity of the efficient coding principle for individual sensory neurons.

For multiple neurons, efficient coding requires that one piece of information is not duplicated in the response properties of more than one neuron. That is, efficient coding is not redundant, but rather reduces the redundancies within the signal (Barlow, 2001). In the visual system redundancies are given by the temporal and spatial correlations in natural scenes (as discussed above), and based on the concept of efficient coding, it has been proposed that the retina and the lateral geniculate nucleus (LGN) already reduce these correlations, effectively whitening the spatial and temporal power spectra (Barlow, 1961; Barlow et al., 1989; Atick and Redlich, 1992; Atick, 1992). Atick and coworkers (Atick and Redlich, 1992) validated the first proposal of a spatial whitening of natural input at the retina, while Dan and colleagues (Dan et al., 1996) showed that LGN temporal response properties are related to a temporal whitening procedure. In summary, the efficient coding hypothesis has been validated at the first stages of visual processing and reduces the spatial and temporal correlations contained within natural scenes. Hence, decorrelation of the signal seems an important characteristic of neural processing.

While neurons in the retina respond to any type of contrast (Barlow, 1961), the responses of neurons in higher cortical areas are specific to contrast of a particular orientation (i.e. oriented edges). Field (Field, 1987) and Daugman (Daugman, 1989) noted that the response distribution of simulated simple cells (cells in V1 with receptive fields shown to be equivalent to localized band pass filters with certain orientations (Hubel and Wiesel, 1962)) are sparse. Mathematically, sparse distributions are defined to be heavier tailed than normal Gaussian distributions. Relating this to neuronal response properties, this means that a single neuron responds only to a particular preferred stimulus. Thus, sparseness is another principle involved in neuronal processing.

Sparseness has been suggested as a general principle to explain sensory neural response properties in terms of reduction of redundancy of sensory input. Olshausen and Field (Olshausen and Field, 1996, 1997) implemented this principle of sparseness as an objective function, which was optimized in a neural network while preserving visual information. The resultant cell population had properties similar to the simple cells in V1. Similar results were observed by using a broader class of Independent Component Analysis (ICA) algorithm on natural images (Bell and Sejnowski, 1997; van Hateren and van der Schaaf, 1998; Lewicki and Olshausen, 1999; Hyvaerinen and Hoyer, 2000). In principle, ICA algorithms decompose the natural images into sparse and statistical independent components. Olshausen and Field (Olshausen and Field, 1997) showed that ICA algorithms are similar to the sparseness approach that they implemented in a neural network. Thus, it has been shown that sparse coding reduces the redundancies contained in natural images. Further work (Hateren and Ruderman, 1998) applied ICA to the spatio-temporal properties of natural stimuli, which led to the derivation of direction-selective cells, again similar to those found in V1, while Caywood (Caywood et al., 2004) successfully applied ICA principles to derive color selective cells. Sparse-coding has also been successfully applied to the auditory system (Lewicki, 2002; Klein et al., 2003). Thus, the simple principles of sparseness and ICA have been used to reduce redundancies in sensory input and are able to explain a variety of different neural properties across different sensory modalities.

1.1.3.2 Temporal coherence

A second generic coding principle, temporal coherence[1], accounts for the invariance of neural responses under transformations of the sensory input. In general, the invariance of neural responses constitutes a loss of information – because the neuron is not sensitive to this kind of change in sensory input, the sensory transformation cannot be detected from the neural response alone. In contrast, sparse coding reduces the redundancy of sensory signals by exploiting the fact that sensory signals are composed of sparse primitives and represents efficient coding without the loss of information contained in the sensory signals. The invariance of neural responses is known to be an important feature of lower and higher cortical regions of the visual system. For example, complex cells in V1 are invariant to the location of sensory stimuli (Hubel and Wiesel, 1962). Another example is the invariance of neural responses in IT (Inferior Temporal Cortex) to rotations of objects (for review Rolls, 2007). The invariance in temporal coherence coding makes use of the fact that features vary on different time scales (Einhäuser et al., 2002). In general, low-level features, like the luminance of a small image region, change on a faster time scale than the relevant variables constituting objects (Körding et al., 2004). Further, object identities vary slower than their position on the observer's retina. In general, temporal coherence keeps the meaningful slow components while discarding the fast-changing parts, and is thus invariant to this fast-changing information. In general the neural response characteristics in higher cortical regions suggest the representation of the world in terms of objects and categories, due to the invariance of neural responses to properties like orientation and location. In this section, we show that temporal coherence has the potential to describe invariance to certain sensory information of neurons in different sensory hierarchies.

As in the case of sparseness, the temporal coherence principle has also been implemented in neural networks. This implementation is similar to employing a temporal low pass filter on the sensory input, and such processing of the sensory input results in shift or phase-invariant filters similar to complex cells (Foldiak, 1991). In addition, the temporal coherence objective has been used to train unsupervised neural networks to learn receptive fields (Einhäuser et al. 2002). The visual input was recorded from a cat behaving in natural environments, and the unsupervised learning process resulted in complex cells. Further, a more analytical approach to temporal coherence, known as Slow Feature Analysis, described the neural response properties of simple and complex cells by using artificial image sequences, generated by moving a window over a natural scene (Wiskott and Sejnowsky, 2002). Shift and phase invariant properties of complex cells were also studied by applying an analytical approach similar to Independent Component Analysis to natural images (Hyvarinen and Hoyer, 2000). In summary, various computational principles have been used to derive shift-invariant properties similar to complex cells from the temporal coherence of natural visual stimuli.

Temporal coherence was also found to be useful in capturing higher-level properties in the visual system, as well as low-level properties in non-visual modalities. Duff and colleagues successfully applied this generic coding principle to the auditory system (Duff et al., 2007).

[1] Temporal coherence is also known as temporal smoothness or temporal stability. Here, I will use only the term 'temporal coherence', but consider the other terms to be equivalent.

Hipp and colleagues (Hipp et al., 2005) applied it to the somatosensory system of rodents, which resulted in representations with higher classification performance of vibrissae sensory data. Similarly Einhäuser and colleagues (Einhäuser et al., 2005) applied the temporal coherence principle to investigate viewpoint-invariant object representation. Other researchers have applied temporal coherence to show translation-, size- and view-invariance of faces and objects (Wallis and Rolls, 1997; Rolls and Milward, 2000; Becker 1999).

Much progress is expected when applying temporal coherence to higher-level representations (Simoncelli and Olshausen, 2001). Wyss and colleagues (Wyss et al., 2006) applied the temporal coherence principle in a convergent, hierarchical network, where levels higher in the network represent higher-order cortical levels, like inferotemporal cortex, hippocampus or entorhinal cortex. The sensory input to the neural network was visual stimuli recorded by a robot moving in an office environment. Sensory input was optimized for temporal coherence at each level of the neural hierarchy, yielding response properties that resemble neurons in the entire ventral visual pathway. For example, on one level neurons were found which vary their firing rate according to the robot's orientation, recalling the head direction cells described by Taube et al. (Taube et al., 1990). On the highest level of the hierarchy, neuronal responses were found to vary as a function of the location of the robot within the environment, properties similar to place cells found in the rat hippocampus and entorhinal cortex (Wyss et al., 2006). Similar results were found by Franzius and colleagues (Franzius et al., 2007), who applied Slow Feature Analysis in a hierarchical structure. Hence, the simple generic principle of temporal coherence can explain complex sensory representation in the entire ventral visual stream.

1.1.3.3 Predictive coding

In recent years the concept of predictive coding has emerged as a solution to perceptual ambiguities contained in natural stimuli. In the first stages of the visual system, stimuli are analyzed locally. However, local information is usually not sufficient to solve the ambiguity related to the reconstruction of 3D environmental features like depth from 2D images (Klette et al., 1998; Ullman, 1979). The human visual system can make use of contextual information, like motion, to resolve these ambiguities (Aloimonos and Shulamm, 1989). To resolve such ambiguities, the theory of predictive coding (König and Krüger, 2006) proposes that the brain anticipates forthcoming sensory stimulation, resulting in a template to which the sensory input can be matched.

Visual representation can be seen as a "prediction error" quantifying the deviation of the bottom-up stimulus from the top-down expectation (Hosoya et al., 2005; Rao and Ballard, 1999). Here, top-down processes refer to neural activity which is transferred from higher cortical hierarchies to lower ones, while bottom-up refers to a transition of neural activation from the sensors to higher cortical areas. For example different groups (Hosoya et al., 2005; Srinivasan et al., 1982) have interpreted the centre and surround characteristics of the receptive fields of retinal ganglia cells as a result of the predictive coding hypothesis. Further, involving the prediction error in neural processing of the sensory input is in agreement with the efficient coding principle, because this error has a smaller dynamic range compared to the procession of the raw sensory input (Barlow, 1961; van Hateren, 1992; Atick and Redlich,

1992). In contrast to the efficient coding hypothesis, predictive coding attempts to not only represent sensory input effectively, but to represent sensory input which can predict future state of the sensory system.

Predictive coding has also been applied to explain a variety of neural response properties. Rao and Ballard (Rao and Ballard, 1999) employed predictive coding in a hierarchical network, with higher levels predicting the activity of lower levels in the hierarchy. In this network, weights between the layers were learned in an unsupervised manner by accounting for prediction error. This unsupervised learning resulted in neural response properties similar to non-classical receptive fields (as characterized by Bolz and Gilbert, 1986) and simple cells. Additionally, different groups (Srinivasan, 1982; Hosoya et al., 2005) have described the surround inhibition of retinal ganglion cells by applying the predictive coding principle. Further, in an fMRI (functional Magnetic Resonance Imaging) studies Summerfield and colleagues (Summerfield et al., 2006) found that a high cortical region, namely the medial frontal cortex, is involved in the predication of the forthcoming perception in a visual classification task. Thus, predictive coding may be a promising theory to further link sensory processing to more and more complex properties of cortical neurons.

Furthermore, it was suggested that optimizing predictability has the potential to result in neural representation similar to symbols and could give new insights in higher cognitive processes. Symbols and categories play a prominent role in capturing and modeling of higher cognitive processes, like perceptual learning (Ashby and Maddox, 2005). Koenig and Krueger (König and Krüger, 2006) argue, that the neuronal representation within each cortical hierarchy can be thought to be the result of the same optimization process, namely optimization of predictability, applied to the input from lower cortical hierarchies.

How is this predictability related to temporal coherence? Temporal coherence extracts features from sensory input which are stable on a certain timescale. This property of temporal coherence can be reformulated: temporal coherence extracts temporally stable features which are predictive in the sense that they do not change. As outlined by König and Krüger (König and Krüger, 2006), stability is the 0th order approximation of predictability, and thus predictability can be seen as a more general principle than temporal coherence. Based on the information theoretic formulation of Bialek and colleagues (Bialek et al., 2001), Creutzig and Sprekeler (Creutzig and Sprekeler, 2008) investigated the relation between temporal coherence and predictability. In the information theoretic framework, predictive coding states that an organism extracts information from its sensory system in order to predict future sensory states (Bialek et al., 2001). This formulation is similar to the so-called bottleneck approach (Tishby et al., 1999), in which the neuron has to encode the sensory input as information efficiently as possible while maximizing the information about future neuronal state. Both Bialek and colleagues and Tishby and colleagues state that “nonpredictive information is useless for the organism” (Tishby et al., 1999). Further, in a first approximation it has also been confirmed that predictive coding contains the solutions of Slow Feature Analysis and thus of temporal coherence (Creutzig and Sprekeler, 2008). Thus, predictability is suggested to entail temporal coherence.

1.2 Combining sensory input and behavior

Sensory optimization has been successfully applied to passively perceived sensory input. The principle of all criteria used in sensory coding is given by imposing normative constraints on all possible sensory representations, i.e. on assumptions about what is considered to be a good sensory representation. Efficient or sparse coding, as shown above, are based on information theory, while stability makes use of the slow changes of visible objects in natural visual stimuli, and predictive coding extracts useful information in order to predict future state of the sensory system. Common to all these principles is that a good sensory representation should represent what is *relevant* (Barlow, 1961). Relevance, however, is beyond the scope of passively perceived sensory input. Rather, relevance is grounded in behavior of an agent that actively interacts with its environment. This highlights the importance of the *sensorimotor space*, spanned by the sensory input and the agent's motor commands. Hence optimal coding of sensory input statistics is not sufficient to effectively code stimuli that are relevant to an organism; sensory input rather has to be combined with the motor action of the agent.

Interestingly, the importance of the connection between motor actions and sensory input has already been emphasized in order to understand perceptual processing (Held and Hein, 1963). More recently, this approach has enjoyed a revival, with O'Reagan and Noe (O'Reagan and Noe, 2001) suggesting that sensory changes induced by actions should influence perceptual organization. They hypothesize that the perception of objects is constituted by predictions about systematic changes of sensory input induced by actions, defined as *sensory contingencies*. For example, learning sensory contingencies in grasping movements of objects can help perception to infer the shape of objects (Hoffmann et al., 2005; Hoffmann, 2007). Empirical support for sensorimotor contingencies is provided by experiments on sensory substitution, which translated visual stimuli into haptic or auditory information (Bach-y-Rita, 1972, 2004) or by transformation of the allocentric orientation of an agent into vibrotactile stimuli (Nagel et al., 2005, 2006). Action-based theories of perception and cognition have also been established in other areas of psychology (Hommel et al., 2001) and philosophy (Grush et al., 2004), as well as the neurosciences. Much of the work on the relation between motor-action and sensory input also has neuroscientific roots in the work of von Helmholtz, who suggested that an efference copy of motor commands could be used to predict the sensory feedback generated by one's own motor action (von Helmholtz, 1867). He further showed that such a mechanism would be useful to remove saccade-related components from the retinal flow field to better detect truly externally-generated optic flow. Recently, Wolpert and Ghahramani applied a Bayesian approach to this idea of an efference copy in order to develop a general theory of motor control (Wolpert and Ghahramani, 2000). Motor processing, in their view, links action and perception, in that it involves a forward internal model that can predict future sensory outcomes based on the current sensory state and motor command. Their implementation of an internal forward model in the motor system is similar to implementations of predictive coding of sensory input, but in contrast, here prediction explicitly takes the motor command into account. Such a direct link between the motor system and perception has recently been reported in an influential neuroimaging study by Naito and colleagues (Naito et al., 2002), who found that the primary motor cortex itself

contributes to the processing of sensory information and the perceptions of limb movements. Also much of the work in the visual system has accumulated strong evidence that sensory and non-sensory information from central sensory visual and motor areas are integrated in order to guide perceptual organization (Leopold and Logothesis, 1999). In summary it can be assumed that the integration of sensory and non-sensory information in sensorimotor space will play a crucial role in our future understanding of neuronal coding properties.

It has been further shown that organisms can extract fundamental properties of the sensorimotor space without previous knowledge of its regularities (Philipona et al., 2003, 2004). As shown in sensory coding, regularities of the sensory input are crucial for the development of neural properties, which can be learned in an unsupervised manner by applying different optimality criteria, like sparseness, to the sensory input. The application of these principles results in a reorganization of the sensory space. Extending sensory coding principles to include the systematic relation between actions and sensory information in the sensorimotor space will provide viable insights into the neural properties and the integration of behavior and sensation.

1.2.1 Introduction of the reorganization algorithm

Here, we aim to use optimal coding schemes to link perception and action. We reorganize the sensorimotor space of an artificial agent by optimizing the predictability of the sensory outcome of the agent's motor commands. Here the sensory space is divided into discrete states. By randomly executing the motor commands available to it and observing the resulting sensory input, the agent learns a forward internal model, based on the sensory outcome of its actions. The forward model was then evaluated with respect to the criteria of predictability and decorrelation. The degree of decorrelation describes the extent to which similar actions resulted in different sensory outcomes, that is, the degree to which actions have different effects. We then applied an algorithm that increases predictability and degree of decorrelation by iteratively modifying the current discretisation of the sensory space. For computational simplicity, this optimization process is guided by a number of rule-based heuristics instead of the more commonly used gradient of an objective function. The sensory space of the agent is given by its 2D position, and the motor space, i.e. the motor commands available to our agent are defined within the ecological niche of a navigational paradigm which is introduced in the next sections.

1.3 Computational models of behavior

Our aim is to apply the optimal coding criterion of predictability to the sensorimotor space, that is, to use predictability to reorganize the sensorimotor space of an agent in the context of navigation behavior. Navigation involves the fundamental processes of behavior: sensory processing, exploration of the environment, and decision making. Here we exemplarily combined these processes in a model of a navigating agent. We claim that this model can be canonically expanded to model other types of behavior. To this end, computational behavioral principles that are already established in neuroscience are introduced in the following sections.

1.3.1 General concepts of behavior

An increasing number of studies model animal behavior using robots. Modeling behavior is the subject of scientific fields as broad as machine learning, artificial intelligence, or biological inspired robotics (Edelman, 2007; Alexander and Sporns, 2002; a discussion of biological inspired robotics can be found in Webb, 2001). Most of this work develops models for a particular behavioral domain, such as navigation or reaching movements. Similarly, in the context of biological inspired robotics, many studies also investigate only an individual component of behavior, such as sensory processing (Lungarella et al., 2003). Although a lot of progress has been achieved within each specific domain, it is becoming more and more obvious that the flexibility of human and animal behavior is still out of reach of modeling studies (Todorov, 2004; Flash and Sejnowski, 2001). As a consequence, recent neuroscientific approaches try to delineate behavior in unified theories that have the potential to span across various specific behavioral paradigms (Wolpert and Ghahramani, 2000; Schaal and Schweighofer, 2005).

In general, an agent performs a behavior in order to reach a certain goal, which can be defined as a desired sensory input. A larger variety of studies (Wolpert and Gharamani, 2000; Flanagan and Wing, 1997) have applied two kinds of transformations necessary for behavior, which are assumed to be applied by the central nervous system. One transformation aims to learn the sensory consequences of executed actions: the forward internal model. The second transformation is based on the internal model and tries to answer the question of which action has to be executed to perceive a desired sensory input: the inverse internal model (for review Wolpert and Ghahramani, 2000). General models of behavior try to provide domain-independent mechanisms to learn both kinds of internal models.

Many studies apply probabilistic approaches to learn a forward internal model because of the noise that is unavoidably contained in sensory and motor systems (Körding and Wolpert, 2004, 2006). A forward internal model aims to describe the sensory outcome of the action, i.e. the transition between the current sensory state and the following sensory state after an action was executed. Within a probabilistic approach, a forward model is thus a set of transition probabilities between two different sensory states related to an action.

1.3.2 Sensory representation

Raw sensory spaces are extremely high-dimensional, i.e. have as many input dimensions as there are receptors. Thus learning forward transition probabilities in the raw sensor space would require enormous amounts of time and storage capacities. As noted by Bellman (Bellman, 1957), the storage capacities required for forward transition probabilities increase exponentially with the number of dimensions, a problem known as the “curse of dimensionality”. Further, increasing the dimensionality of the sensory space increases the exploration time needed to map forward transition probabilities by explorative sampling. However, generic coding principles applied to the sensory space yield low dimensional sensory representations which show an enormous potential in reducing the sensory curse of dimensionality. Several groups (Franzius et al., 2007; Wyss et al., 2006) have recently shown that the optimally stable representation of a behaving robot’s visual input is not only low dimensional, but also matches the long-standing experimental reports of place cells found in

rodent hippocampus (O'Keefe and Dostrovsky, 1971). These neurons preferentially fire when the animal is located in a certain region of the environment, defining the cell's place field. As shown by Wilson and McNaughton (Wilson and McNaughton, 1993), with the help of the activity of different place cells the current position of the rodent can be encoded[2]. Furthermore, Morris and colleagues (Morris et al., 1982) showed that place cells contribute to a rodent's navigational ability within mazes, suggesting that place cells constitute a neural map of the spatial environment (O'Keefe and Dostrovsky, 1971; O'Keefe and Nadel, 1978; McNaughton et al., 1996; Samsonovich and McNaughton, 1997). Here we continue the work of Wyss and coworkers on unsupervised learning of visual input to organize the visual input space into states that have spatial meaning about the position of the agent in the world. We use place cells as a low-dimensional sensory representation that can locate a robot in its environment, and learn a forward model for the action consequences involved in navigation by means of state-to-state transition probabilities between place cell activations.

1.3.3 Actions and motor primitives

Up to now we built the forward internal model using the behaviorally relevant low-dimensional sensory representation of place fields. The forward internal model captures the sensory consequences of the agent's actions. In this study, we implemented these sensory consequences, which are represented as state transitions, as transition probabilities. These transition probabilities represent the internal forward model of the agent and contain the knowledge of the agent about the world.

To learn forward models, we also have to consider that a particular behavior, for example reaching for a cup, can be performed in many different ways. The number of joints and muscles in our skeletons also suggest a high dimensional space of motor capabilities. In recent years, neuroscientists faced this second problem of dimensionality using the notion of "*motor primitive*", suggesting that complex behaviors are built from primitive motor units by sequential or simultaneous combination (for definition see Flash and Hochner, 2005). These units can be investigated on a neural level as well as on a kinematic and dynamic level. They are non-redundant modules and thus alleviate the curse of dimensionality in understanding behavior. A simultaneous or sequential combination of these motor primitives results in a certain behavior and thus can capture all possible different possibilities the behavior can be performed.

For reasons of simplicity, here we restrict actions to motor primitives and their sequential combination.

1.3.4 Combining actions to behavior

Accomplishing a certain task requires finding suitable actions to reach the desired goal, which is in general done by satisfying optimality criteria. As goals are defined by a desired sensory input, we can use inverse internal models to determine which actions have to be

[2] It should be noted that the activity of place cells is also sensitive to non-spatial sensory properties (Eichenbaum et al., 1999), to events associated with the place (Markus et al., 1995; Moita et al., 2003) and to events associated with the navigational goal (Hok et al., 2007). However, visual input is necessary for the formation of place cells (O'Keefe and Burgess, 1996), as modeled by Wyss and colleagues (Wyss et al., 2006) as well as Franzius and colleagues (Franzius et al., 2007).

executed in order to perceive the desired sensory outcome. But there are numerous possibilities to combine the different actions in order to obtain a particular goal, which are qualitatively different according to different optimality criteria maximizing the potential reward, precision and smoothness (for review see Todorov, 2004). Finding an inverse internal model is thus an optimization problem as formulated in optimal control theory (outlined by Bellman, 1957). As a standardized method, especially in reinforcement learning (Sutton and Barto, 1998), the action which maximizes the reward is chosen for execution. Thus, based on the internal forward model, an evaluation must be made of which action results in a maximal reward.

1.3.5 Introduction of behavioral model

To model navigation in an environment, the architecture of our agent distinguishes central and distal processing. Both processes learn about sensory outcomes of the robot's actions in the agent's state space, which is spanned by the low-dimensional sensory representation of place fields. Central processing components learn sensory outcomes of the agent's actions in the form of a forward internal model of probabilistic state transitions. In contrast, the distal component accounts for the use of distal sensors, such as infrared sensors, driving a reflexive obstacle avoidance behavior. The distal processing component memorizes the occurrence of obstacle avoidance behavior during a state transition as a so-called reflex factor. The transition probabilities and the reflex factors reflect the environmental properties in relation to the robot's actions. Thus, by random action execution, the robot can learn an approximation of the environmental affordances (Gibson, 1977) for navigation, defined as the navigational action possibilities afforded by the environment. The cognitive model plans goal-directed actions by integrating the information gained by central and distal processing, i.e. goal-directed and reflexive behavior, into a local decision-making process which maximizes the reward. This integration results in a quantitative measure of how reliably each executable action leads towards the goal. In summary, the key components of our cognitive model are (i) a high-level representation of sensory input space (place fields), (ii) the knowledge of environmental properties acquired by active exploration of local state transitions by means of distal and central processing and (iii) a decision-making process driven by both types of knowledge.

2. Computational model of behavior

2.1 Method

2.1.1 Agent Architecture

Our cognitive model allows an agent to experience the environment and navigate to different targets based on a state space represented by the spatial representation of place fields. Here the agent controls a Khepera robot (Fig. 1C) to navigate in a four-arm-maze environment (Fig. 1A), consisting of platform with four connected arms, a standard environment, developed by Olton and Samuelson (Olten and Samuelson, 1976) to investigate spatial learning and memory of rats. The state space was obtained by dividing the four-arm-maze environment (Fig. 1A) into compact, discrete states (Fig. 1B), similar to the place fields that

can be acquired by unsupervised learning. The central component of the model processes every one of the robot's state transitions, while the distal component deals only with transitions that coincided with reflexive behavior (obstacle avoidance). Together, the transitions induced by the robot's actions and those transitions associated with reflexive behavior represent the environmental properties locally learned by the exploring robot. During each stage of the decision-making process, the model chooses the action that maximally increases the probability of reaching a desired target within the environment, thus allowing the robot to successfully navigate.

2.1.1.1 Sensory processing

We chose place cells as a representation of the environment. A study by Wyss and colleagues (Wyss et al., 2006) showed that such place cell properties can be acquired by mobile robots by means of unsupervised learning in a hierarchical network. Although it would be possible to replicate this work, we prefer to build on these previous results and extend that study. Our main purpose here is to model behavior, so we deliberately used predefined place cells to simplify the task. We approximated the firing properties of place cells as a function of the robot's position by 2-dimensional Gaussian functions (standard deviation: 0.04 m). To cover the whole four-arm-maze environment we randomly distributed 72 of these Gaussian functions (Figure 1B). For each of the robot's possible positions within the maze, we obtained the activity of each of these place cells. A winner-takes-all process then extracted the robot's position in state space from the population activity of the place cells – the cell that was maximally active thus defined the current state of the agent. In order to calculate the place cell activity, we first needed to extract its position in the environment. The robot was tracked by an Analog Camera (Color Cmos Camera 905C), which was attached above the environment as shown in Figure 1A. The analog camera signal was digitized by a TV card (Hauppauge WinTV Express). The position and orientation of the robot were then calculated using the camera image and the color code attached on top of the robot. Thus, the population of place cells represents a mapping from the position space in which the robot is navigating, to the state space of the agent controlling the robot. To secure later generalization, no reference of any kind is made to the 2-dimensional structure of the environment. The only information used by the agent to infer the robot's position is the activation of each place cell.

2.1.1.2 Action execution

In order to limit the number of transitions needed to learn the environmental properties to a manageable number, in each state the robot was restricted to executing eight different actions. Each of these actions consisted of a static rotation to a certain orientation followed by the straight-line movement of the robot. The corresponding orientations were equally spaced from 0 to 360 degrees. As a result of executing such an action when in a given state (source), the robot will reach a different state (end state), with the action thus resulting in a *transition* between states. An end state is reached when the winner-take-all process calculating the current state returns a new state index. A transition is defined as complete when a local maximum of the end state's activity is reached. A local maximum occurs when the derivative of the current state's activity becomes negative. The frequencies of the transitions resulting

from action i, executed in source state j and ending in end state k are stored in the *experience matrix* $EM_{i,j,k}$.

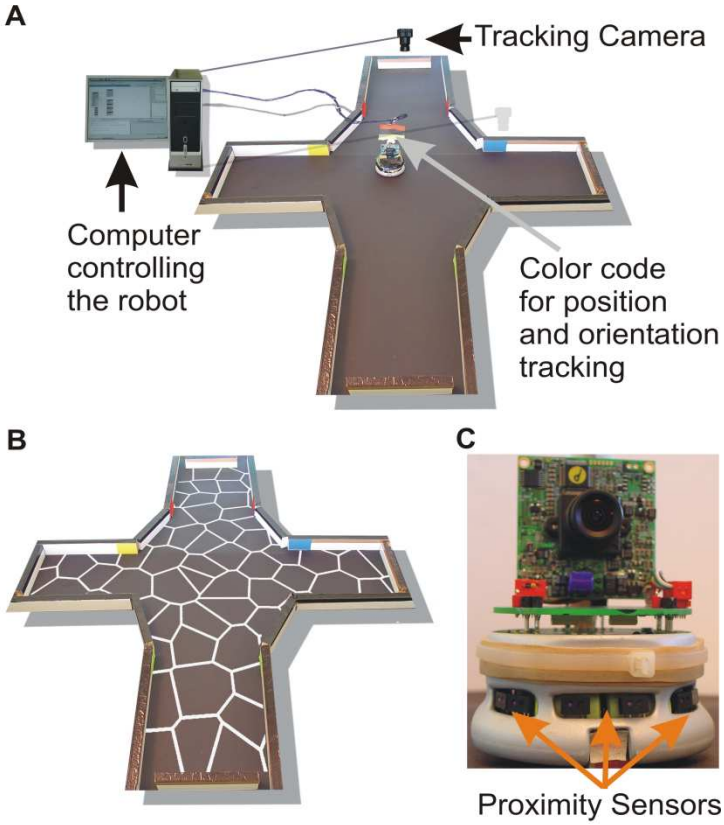

Figure 1: (A) A four-arm-maze environment was chosen to test the model. A Khepera robot was controlled by an agent implemented in the MicroPsi framework, running on a computer. The inputs to the program were the orientation and the position of the robot, as well as the data from the proximity sensors. Part B shows the distributions of states used. The white lines within the environment represent the boundaries of these states. C shows these proximity sensors on the robot. These sensors emit infrared light and measure the reflection of this light at obstacles. The agent used the activation of the place cells which corresponded to its current position, the orientation of the robot, and the proximity sensors to perform the robot's behavior.

2.1.1.3 Distal processing

To prevent the robot hitting one of the maze's boundary walls, a reflexive obstacle avoidance behavior was implemented. The proximity sensors (Fig 1C) were used to perform this behavior. The frequencies of occurrence of the reflexive event characterized by the particular state (i) – action (j) combination is stored in the *reflex matrix* $RM_{i,j}$. Whenever the robot used its obstacle avoidance behavior, the system associated the current state and action with the occurrence of a reflex event and updated the reflex matrix accordingly.

2.1.1.4 Decision making

The properties of the environment (boundaries, obstacles, etc.) determine how likely it is that a certain state transition will occur given a chosen action. These transition probabilities are approximated and learned by the agent as it explores its maze environment and are stored in a *transition matrix* (Figure 2A). The 3D transition matrix consists of a 2D matrix for each action i TM_i. The row index determines the source state j and the column index represents the

end state k of this action. Thus the transition probability defined by source j, end state k and action i is stored in the transition matrix $TM_{i,j,k}$ shown in Figure 2A. The sum of the transition matrix over the end states k (rows) is normalized to one for each action and source state and thus represents a probability distribution. In the experiments described below, the robot learned the transition probabilities are based on 240 minutes of random maze exploration.

Next we address the problem of decision-making – choosing the action that is most likely and quickest to lead to the goal. To accomplish this, an iterative *reverse flooding* approach was introduced, which integrates the environmental properties represented by the transitions and reflex factors (Fig. 2B). The properties learned by the central component of the model are stored in the 8 transition matrices TM_i, and share similarities with a directed graph. The vertices of this graph correspond to the states, the edges to the transitions, and the edge weights to the transition probabilities. This results in 8 directed graphs equivalent to the eight possible actions. In each of the iteration steps of reverse flooding, the activation of the state corresponding to the goal state is set to 1. State activation is propagated through the graph by passing the activity – weighted by the corresponding transition probability – to connected states in the *reverse* direction of the directed edges. Technically speaking, the activation is propagated from end states to sources, weighted by the transition probability of the action's transfer from the source to the end state, hence the name reverse flooding. Applying this process to each action's graph gives rise to 8 different activity values for each state. Up to this point, only the learned environmental properties resulting from central processing have been considered during flooding.

To further integrate the learned properties caused by distal processing, we introduced reflex factors. The reflex factor is proportional to the percentage of actions i at source j that had an associated reflexive event:

$$rf_{i,j} = 1 - \left(\frac{RM_{i,j}}{\sum_k EM_{i,j,k}} \right) \cdot \frac{5}{6}$$

RM and EM represent the reflex and the experience matrix. During each iteration step of reverse flooding, the eight activations of state j corresponding each to one of the eight actions i were multiplied by the corresponding reflex factor $rf_{i,j}$. The maximum of the eight activations of a state was used as the state's activation for the next iteration step. This iterative process was continued until the states' activities converged. In order to select the action most likely to move the robot towards the goal, we considered the eight incoming activation values on each state, which resulted from the activation propagation of the eight actions. The robot then chooses the action corresponding to the highest incoming activation of the current state.

For the computation of the reflex factors we weighted the ratio of reflexive transitions to all over transitions between two states by a factor of 5/6. This weighting prevented the reflex factor of being zero in the case of an action which is combined only with obstacle avoidance behavior. Further, nonzero reflex factors did not neglect the information of the environment

gained by the transition probabilities during the flooding process by setting the activation to zero.

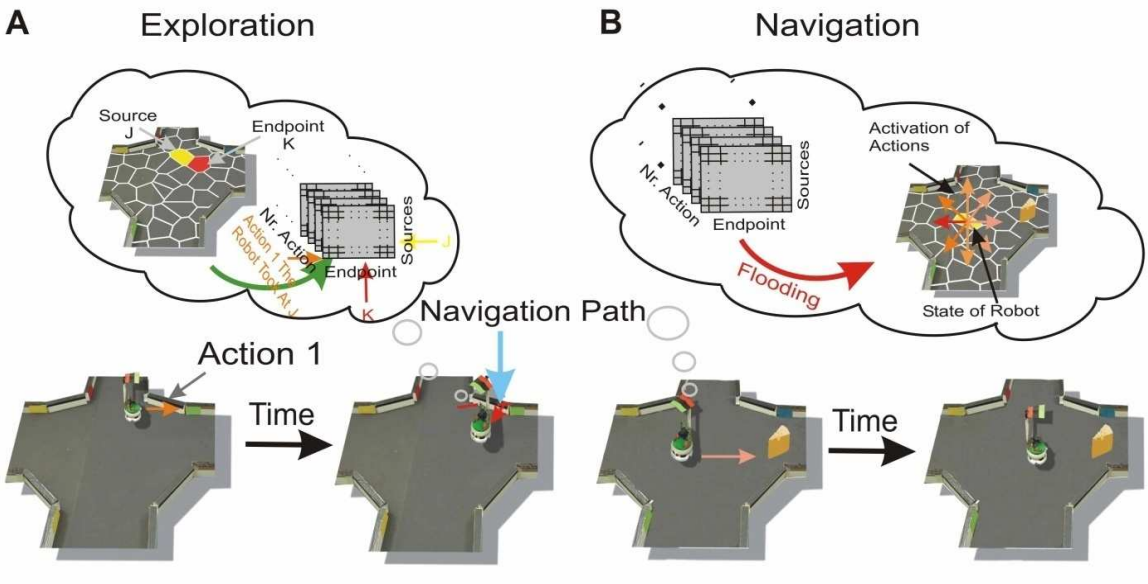

Figure 2: (A) Learning of the properties of the environment. The robot is on a certain state, defined here as Source J (yellow labeled) and randomly chooses an action (Action 1). The execution of the action results in another state, defined as end state K (red labeled). This transition was stored in a three dimensional matrix, called the experience matrix, with the dimensions sources, endpoints and actions. The number of action executions combined with obstacle avoidance from a source was stored separately. (B) The robot moving to a goal (the "cheese" for the artificial "rodent"). His choice is a consequence of the flooding of the transition matrix, resulting in an activation of the different actions, shown as colored arrows. The action with the strongest activation was chosen.

Furthermore, we introduced a *decay factor* df, which was here set to 0.9. After each iteration step, the activation of each state was multiplied by this factor. The more transitions that are needed to reach the goal states, the more the decay factor is taken into account and decreases the states' activities. Hence, the decay factor penalized longer trajectories to the goal state.

The flooding algorithm defined above was implemented with the help of matrices.

$$act_j^{(0)} \begin{cases} 0 & j \neq m \\ 1 & j = m \end{cases}$$

represented the activation at the 0'th activation propagation, where the goal was located at state m.

$$\overrightarrow{act}(t+1) = \max_i((TM_i \cdot \overrightarrow{act}(t)) \cdot rf) \cdot df + \overrightarrow{act}(0)$$

where $\overrightarrow{act}(t)$ is the vector of activation values for the states after t iteration steps. rf represents the reflex factor and df the decay factor.

2.1.2 Robot setup

To test the model in a real-world environment we used Khepera II robots (K-Team, Lausanne, Switzerland). The robot was equipped with 8 proximity sensors, which emitted infrared light and measured the strength of its reflection, and two wheels, each controlled by one motor (Fig. 1C). For implementation and flexible programming, we used MicroPsi (Bach, 2003; Bach and Vuine, 2003), an Eclipse-based Java programming environment, as an interface to the robot. The agent that controlled the robot's behavior was implemented in this framework. The real-world environment was a four-arm maze with boundaries built from white wooden pieces (Fig. 1B). Each arm had a width of 0.21 m and a length of 0.28 m. The four-arm maze environment fitted into an area of 1 m^2.

2.1.3 Methods of analysis

As a means of comparison, a simulated robot was implemented in MATLAB (Version 7.0 (R14), Mathworks, Natick, MA, USA) using the same algorithms described above. For this simulation, we chose the same environment with the same dimensions like the robot was situated in. Obstacle avoidance behavior was implemented by setting the angle of reflection equal to the angle of incidence to the boundary, with a random scatter of 10 to –10 degrees added.

To compare the navigational behavior and the transition probabilities learned by the robot, we introduced the *geometrical transition matrix*. This matrix takes into account only the topographical properties of states in the environment and was created by allowing the simulated robot to execute every action on every position within each state, using the resolution of the camera tracking system. Because the real-world robot chose a new action only at a local maximum of its current place cell activity, each transition occurrence in the simulated agent was weighted by the probability of the robot executing an action given the current place cell activity. In an ideal world and given a very long exploration time the real transition matrix is expected to converge to the geometrical transition matrix. In the real world setup, due to the finite robot size, slip and friction and a limited exploration time, the geometrical transition matrix might deviate considerable from the real transition matrix.

Next we evaluated the properties of the experienced and geometrical transition matrices. First we investigated the *similarity of action outcomes* by comparing the corresponding transition probabilities. We correlated the transition probabilities represented by a row vector of the Transition matrix of action i, TM_i, with the same row vector of the Transition matrix of action j TM_j. Before calculating the correlation coefficients between the two vectors we reduced the transition probabilities in the row vector by the average of these transition probabilities to the topographical next neighbors. Thus two actions lead to equivalent outcomes when their correlation coefficient is 1.0; they are linearly uncorrelated when the correlation coefficient is 0.0. Thus the correlation between the transition probabilities from state i and the probabilities from state j of action k is defined by:

$$corr_{i,j,k} = \frac{\sum_l \widetilde{TM}_{i,l,k} \cdot \widetilde{TM}_{j,l,k}}{\sqrt{\sum_l (\widetilde{TM}_{i,l,k})^2 \cdot \sum_l (\widetilde{TM}_{j,l,k})^2}}$$

$\widetilde{TM}_{i,l,k}$ is the transition probability between state i and l with action k, reduced by the average transition probability to the topographical next neighbors of state i.

We characterized the predictability of an actions' transition to a state by defining a second measure: The *predictability* of action i in state j is given by the maximum transition probability stored in the row vector j of the Transition Matrix TM_i. This maximum transition probability was reduced by the probability of transferring to one of the connected states by chance.

$$Pr_{i,j} = max_k\left(TM_{i,j,k}\right) - \frac{1}{conn_{i,j}}$$

Here, $Pr_{i,j}$ corresponds to the predictability of action i in state j, and $conn_{i,j}$ is the number of states the robot can reach by executing action i on state j.

In order to evaluate the decision-making process, we analyzed the activation of each action after the flooding process had converged. We chose the *normalized activity* as an appropriate measure to characterize the strength of selection of an action during navigation to a goal. This activity is defined as the activation of the chosen action for the state, normalized by the sum of all incoming activity and by the decay factor.

$$NormAct_j = \frac{\overrightarrow{act_j}}{\left(\sum_i((\mathrm{TM_i} \cdot \overrightarrow{\mathrm{act}}(\mathrm{t})) \cdot \mathrm{rf}) \cdot \mathrm{df} + \overrightarrow{\mathrm{act}}(0))_j\right) \cdot (1 - df)}$$

with $\overrightarrow{act_j}$ representing the maximum converged activity of state j after flooding. The denominator corresponds to the sum of all converged activations of state j over all actions. In order to reduce the dependency of the normalized activation on the decay factor, we included the decay factor in the denominator. As a result of its inclusion, the normalized activity ranged from 0 to 10.

2.1.4 Batch and online learning

We investigated the plasticity of the introduced navigational system by examining the robot's navigational adaptation to changes in the environment. The robot's navigational performance was evaluated by measuring its path to a target. We examined the robot's adaptation process by comparison of the two different types of learning we introduced: *batch* and *online learning.* These approaches differ in the timing of the transition matrix and reflex update and in the way in which the robot explores the environment. Batch learning involves interleaved experience stages of random action execution, during which the existing transition and reflex matrices are updated, and evaluation stages, during which navigation takes place and the transition and reflex matrices are not updated. This is similar to the way the agent experienced the environment as described above. In comparison, online learning involves updating the robot's transition probabilities and reflexes after each action execution, and instead of moving randomly, the decision-making process is always at work.

2.1.5 Robot control

2.1.5.1 MicroPsi framework

The purpose of the developer of MicroPsi was to formalize and implement the Psi Theory of Dörner (Dörner, 1999). The Psi theory addresses emotion, perception, representation and bounded rationality, in order to capture different components of situated agent. This theory is formulated within psychology. MicroPsi is a formulation of this theory in an abstract and formal way, in order to build situated agents facing question of attention and ontological categorization. This program provided tools to easily implement agents based on the Psi theory in a simulated environment.

The first task was to adapt MicroPsi in order to control a real robot. The principle modules of the program were the simulated environment (world), the agent framework and the server. The world constitutes the environment were the agent was embedded and were the actions of the agent were executed. This component also generated the perceptions of the agent in the simulated environment and sent them to the agent framework. In the agent framework these perception were processed and action were calculated in order to show a particular behavior. The action commands were sent to the world in order to be executed. The communication between the world and the agent framework was done by the server. We had to exchange the world component by a robot component in order to control the robot and provide the agent framework with the robot's sensory data.

The main part of the robot component consisted of a thread which controlled the communication between the computer and the robot. This thread had to control the sending of the command controlling the speed of the robot's two wheels. Furthermore, it sent the command to read out the proximity sensors and get the answer of this request.

These sensory inputs were given by the position, orientation of the robot and the values of the proximity sensors. In the next section we will discuss how the different sensory data were processed and communicated to the agent framework.

2.1.5.2 Tracking the robot

In order to extract the current position of the robot, we used an overhead camera (shown in Figure 1) and the color code attached at the top of the robot (red/yellow). Every 20 msec an image of the overhead-camera was captured and analyzed the red and yellow colors of the robot's color code. We extracted the yellow/red color of the color label by filtering the images in the HSV color space. Based on these filtered images, we calculated the weighted mean of the pixels, by multiplying the x and y index of the pixel with its activity in each of the color filtered images. This resulted each in a mean x (horizontal) and y (vertical) position of the red/yellow label. The position of the robot was calculated by adding to the mean x and y position of the red label half the vector spanned from red to the yellow position. Thus the position of the robot was the given by the center of the black region which divided the red and the yellow part of the color label (shown in Figure 1). This position was chosen as the one of the robot in order to be closer to the axis of rotation of the robot, because the position of the robot should be invariant under rotation. In order to reduce the computational effort,

each image was only analyzed around a small region of the previous computed position of the robot. This procedure to measure the position of the robot resulted in a stable positioning of the robot with a standard deviation of 1.76 cm under repeated measurements.

The orientation of the robot was extracted by measuring the angle between the vector, spanned from the position of the red to the yellow label, and the horizontal line of the captured image of the overhead camera. This resulted in a stable appointment of the orientation of the robot with a standard deviation of 2.56°.

There are different methods to extract the orientation and position of the robot. One is for example to track the shape of the robot. In general this procedure is implemented by using the OpenCV (Zhang, 2000). At the time we had to implement the behavior of the robot an interface between java and the OpenCV libraries did not exists, and thus we chose the easily implementable procedure to extract the position and orientation of the robot with the help of a color code. We also calibrated the overhead camera utilizing OpenCV so that we did not obtain any distortion from the lenses.

2.1.5.3 Readout of proximity sensors

The proximity sensors are measuring the distances to different objects by measuring the reflection of infrared light, emitted by the same sensors. We implemented a thread which accessed the values of these sensors every 30 msec. This was accomplished by first sending a command to the robot, which then sent these values back to the computer. The values sent from the robot varied between 0 and 1023.

2.1.5.4 Sending motor commands

Each time an action command different to the previous one that arrived from the agent framework at the robot component, the command was sent to the robot. Here the command sending the motor command had a higher priority than the command accessing the proximity sensors.

2.1.5.5 Agent Framework

The main component of the agent framework is the nodespace. Nodes are entities that are characterized by an input and output. These nodes can transform the incoming activity and transfer it via the output to connected nodes. These nodes have different outputs and can be connected via different connections. Different types of nodes and connections are implemented such that any first order logical process can be implemented by a net of nodes.

In order to provide the sensory information to the nodespace we had to implement another component of the agent framework, the agent-world adapter. The agent-world adapter deals with the communication between the nodespace and the robot component. This agent-world adapter provides the sensory data to the nodespace. The nodespace transforms the sensory input in order to obtain a certain behavior and thus calculates the action commands. The action commands were transferred to the robot component by the agent-world adapter and the server. The nodespace updated every 30 msec the sensory data and the actions commands. This 30 msec is also the cycle the activity of one node is transferred to another node.

2.1.5.6 Implementation of the agent

To implement the cognitive agent as described above we only used some of the provided tools. We mainly used native modules, which were nodes containing java code and were able to use all functions accessible to java to transform the incoming activity. The implementation of the agent in the nodespace was done by transforming the sensory activity, given by the x, y position and orientation of the robot as well as the proximity sensors. Here we used only six frontal proximity sensors of the Khepera robot. All the provided sensory activities entered the nodespace by so called sensory nodes as shown in Figure 3. These sensory data were transformed such that the robot shows the navigational behavior as discussed above. This behavior of the robot was controlled by setting the speed of each of the two robot's wheels. This speed of the two wheels was set by the activity of the two Actor nodes as shown in Figure 3. This activity was then sent to the robot.

The procession of the sensory signals was done in two different node-subspaces. One of these node-subspaces (Memory, see Figure 3) contained the elements for the flooding process and calculated the transition probabilities as well as the reflex values. The native module Floodingtest transformed the coordinates of the robots position in the current state and activity of the place cell. In case the conditions to make a decision are fulfilled as described above, this native module sends one of the possible actions to the allocentric native module.

The AllocentricEgocentric native module translated the action parameters into motor commands and adjusts the orientation as well as the speed of the robot's straight movement. This was done by setting the speed of the two wheels of the robot. When a new action was sent to the AllocentricEgocentric native module, the robot started to rotate until the robot adjusted its orientation of the action to be executed. The robot rotated in this direction of the smallest difference between the current orientation and the one which has to be adjusted. Further the AllocentricEgocentric native module had the proximity sensors as input. This input was used to prevent the robot to hit the wall during the adjustment of the orientation of the action. Thus in case the rotation in one direction resulted in an obstacle detection by the sensors, the robot rotated in the other direction. Only if the second direction of rotation resulted also in obstacle recognition, the rotation was stopped and the action to be executed was associated with a reflex and the movement of the robot was controlled by the proximity sensors. In order to account for this fact, the AllocentricEgocentric native module was connected to the native modules in the Braitenberg sub-nodespace.

The node-subspace (Braitenberg) controlled the reflexive behavior in order to prevent the robot from hitting the wall. This subspace consisted of 9 native modules, as shown in Figure 3. 6 of the native modules (SensorScaling) normalized the raw data of the proximity sensors ranges from -1 to 1. The lowest value is due to a position of the sensor very close to the wall and 1 for sensing no obstacle. The native module SensorAdaptation low passes the values of the proximity sensors. The FilterMotor module projected the values of the filtered proximity sensors to the speed values of the robot's wheels. This native module further controls that in case an obstacle is detection is thwarted the adjustment of an orientation or an obstacle is detected while a robots straight movement, the proximity values set the motor output. Otherwise the AllocentricEgocentric module sets the motor output. The ReflexData native

module detects if a reflexive event occurred and send it to the central Floodingtest module of the Memory subspace.

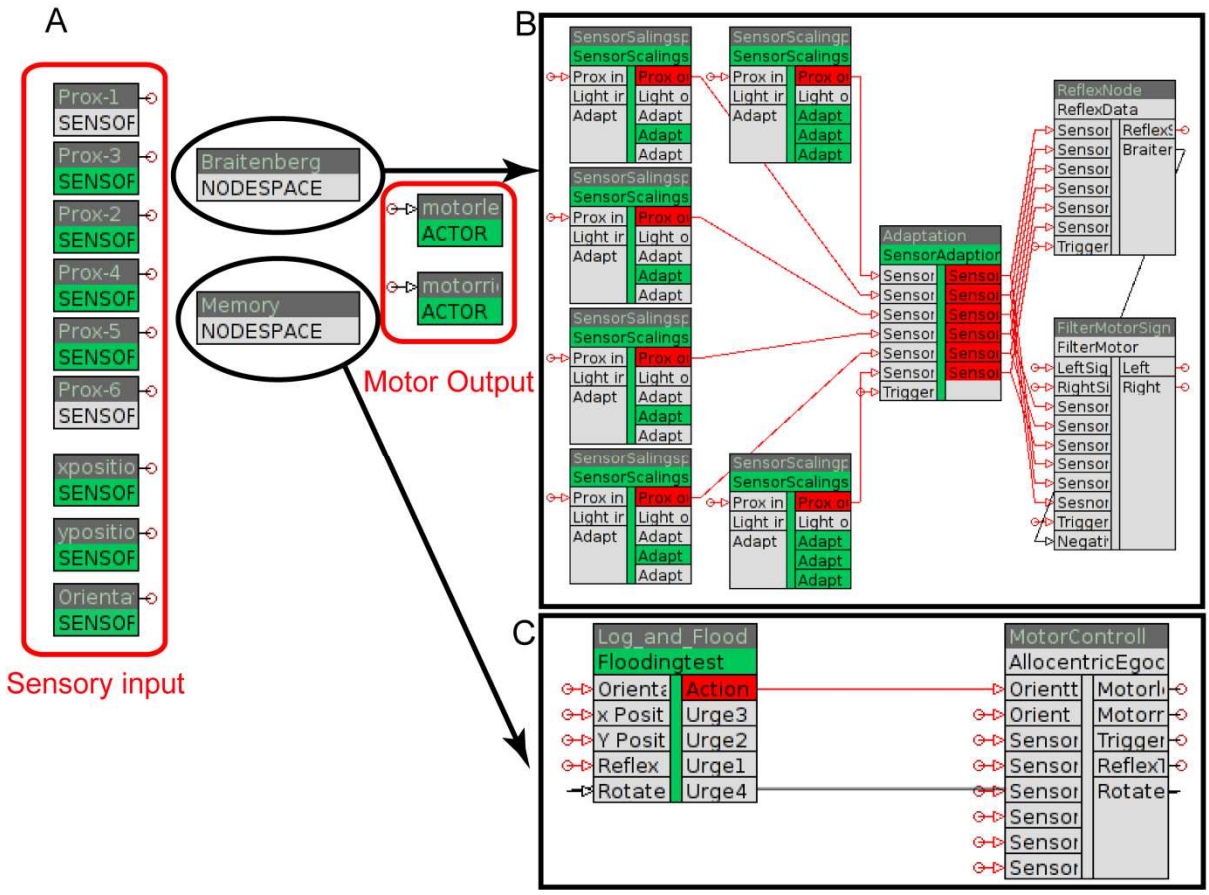

Figure 3: Structure of the nodespace. A: In the nodespace the sensory input (6 proximity sensors and x/y position and orientation) is transformed in the two sub-nodespaces into a motor output, which set the speed of each wheel of the robot. B. The Braitenberg sub-nodespcace controlled the reflexive obstacle avoidance behavior. The SensoryScaling nodes processed the activity of the proximity sensors and normalized the values between -1 and 1. Here -1 corresponds to an obstacle in front of a proximity sensor and 1 to no obstacle detected. In the SensorAdaptation native module we low passed the proximity sensors. The ReflexData module controls if a certain reflex occurred, while the FilterMotor maps the activity of the proximity sensors to the speed of the robot's wheels. This module controls the motor activation by setting the speed using the sensory values or the action commands controlled by the agent. C. The Floodingtest is the central modul, which takes the decision of the action to be taken by the robot, stores the transition probabilities, calculates the flooding process and transforms the x and y coordinate to the robots state. The different action commands were sent to the AllocentricEgocentric module which translates the action commands to the corresponding speed of each wheel.

2.1.6 Four-Arm-Maze task

The Four-Arm-Maze task is a standardized experiment used in experimental work with rodents (Olton and Samuelsen, 1976). In this setup, the rat learns to associate a sensory input with a reward in the future, at a certain location in one of the arms. In this experiment a reward is located in one arm, which is always labeled with the same sensory stimuli. Changing the labels of the arm and also the reward location, the animal should only enter this arm, which has the reward location. In most experiments a tactile input is used, which can be

discriminated by the rodent's whisker system. In order to simplify the experiment we labeled the different arms with visual cues as seen in Figure 1A. These cues can only be discriminated in a small area around the cues, similar to the tactile stimuli.

2.1.6.1 Sensory processing

In order to provide the agent with the visual stimuli of the labels we used the frontal camera of the robot as shown in figure 1C. The images provided by this camera were grabbed and digitalized by WINTV EXPRESS CARD. We captured each 20 msec one image with a resolution of 320x240. For each pixel the activity for four different color regions in hsv color space was analyzed. The color regions were adapted to the colors of the labels. These activities were summed over all pixels for each of the color regions resulting in the labels activity of the current image. We further provided the agent with so called color sensors. These color sensors were set to one if the labels activity was higher than a certain threshold and zero else. Thus these color sensors were one only in a small region around the label.

2.1.6.2 Modification of the agent

Similar to the rat in the learning phase, the agent had 4 different possible reward states, one in each arm. But in each trial the reward is located in only one arm labeled with particular visual stimuli, which was here a blue label. At the beginning of the trial the agent executed a flooding process from these four reward states (activation was set to 1 on each reward state), which resulted in a movement towards one of the four arms. At the entrance of the arms the agent had a visual sensory input, given by the labels of this arm. At this location the robot has to make a decision in order to either move in the arm, if it's a blue label or not. This procedure corresponds within the agent framework to either a reduction of the activation at the corresponding reward position and thus the robot do not move into the arm, or an increase of the activation, such that the robot moves into the arm to find the reward. Here the main problem is which of the four possible reward states are associated with the current visual stimulus at the entrance of the arm.

This problem was solved by a forward flooding process. This process is defined by the propagation of activation through the graph structure of the learned transitions from the state of the agent, a sensory input is perceived. This results in a certain activation of the four different reward states. The most activated reward state is associated with the current stimuli and either more activated (activation was set to 10) in order to get the reward or set to zero to prevent the agent to enter the arm. In case the robot found the right reward state in the right arm, the activation of this reward state was set to zero. After each change of the activation a new flooding procedure, either forward or backward was executed. The task was accomplished in case the activation of all potential reward position was zero.

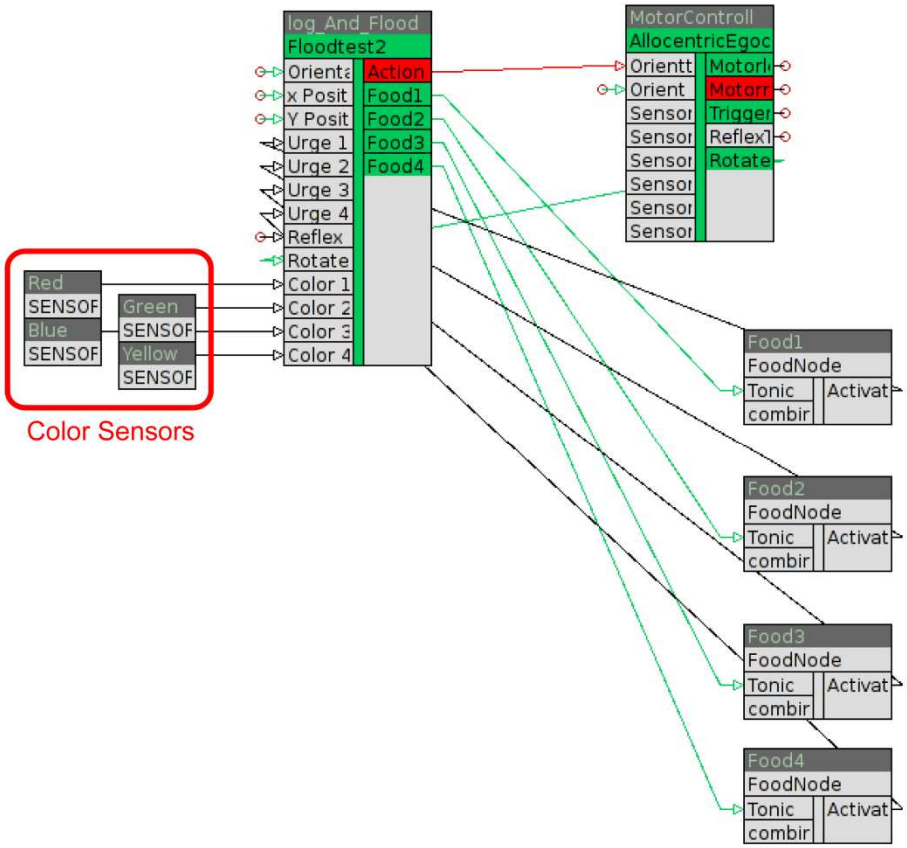

Figure 4: Modification of the sub-nodespace Memory (see Figure 3) in order to accomplish the four arm maze task. The color sensors detected the labels of the different arms, while the FoodNodes controlled the activation of the 4 possible reward states in the different arms of the maze.

2.1.6.3 Implementation of the agent

We expanded the implementation in the agent framework as shown in Figure 3 by different nodes. Here we modified only the Memory subspace. The connection to the other subspaces and the organization of the other nodespaces was the same as shown in Figure 3. First we expanded the Flooding native module by the forward flooding. Further we added color sensors in order to detect the different labels as shown in figure 4. The activities of the states were potentially reward could be found were controlled by the FoodNodes. In case a label was detected by the color sensors, the corresponding reward state was determined by the forward flooding process. In case the label was detected with the reward state in the arm, the corresponding reward state was more activated otherwise the activation was switched off. With the help of these tools we implemented the agent, which is able to successfully complete the four-arm-maze task.

2.2 Results

Here we investigated the robot's navigational performance and how the central processes – namely the transition probabilities – as well as the distal processes defined by the reflex factors, contributed to the decision-making process. We also examined the adaptation of the robot's navigational behavior to changes in the environment.

2.2.1 Navigation Behavior

The navigation performance of the robot was evaluated by repeatedly measuring its path to a number of different target sites in the environment. In each of the 20 trials, the robot was placed on one of five possible starting positions and given one of four target locations. In order to directly compare different start-target combinations, we normalized the length of the robot's path by the *direct path,* which represented the shortest traversable distance from the robot's starting point to the goal state. Figure 5 shows a path traveled by the robot (yellow line) and the corresponding direct path (light gray line). Overall, the robot's median path length across 20 trials was 1.71, with a standard deviation of 0.47. This represents an increase of 71% (± 47%) compared to the direct path length. For all configurations of start positions and targets, the robot was able to reach the target in a reasonably short amount of time.

The increased length of the robot's paths could be a consequence of any of the following: the division of the environment into discrete states (place fields), the environmental properties learned by the robot (transitions and reflex factors), and the robot's behavior while navigating through the environment. Each of these factors was investigated in turn. To provide a first approximation of the increase due to the discretization of the environment, we simulated the robot's behavior using the same navigational algorithm as described in the Methods section. The simulation used the geometrical transition matrix, which takes only the topography of states into account (see 2.1.2), to navigate from the same start positions to the same goal states as the real robot. The red line in Figure 5 shows a sample path of the simulated robot. This simulation resulted in a median increase of 19% (±9%) compared to the direct path. Thus, the introduction of discrete states did not greatly contribute to the lengthening of the robot's path to a goal.

Next we investigated the contribution of the robot's learned environmental properties. To do so, we again used the simulated robot with the same start-target combinations, but this time used the robot's learned transition matrix and reflex factors to perform the task. Figure 5 shows an example of such a simulated path (green line). The median increase in path length was 37% (±23%). As 19 % of the path increase is caused by discrete states, approximately 18 % is due to differences between the geometrical properties of the environment and those properties learned by the robot. Thus, the difference between the geometric and learned transition matrices and reflexes explains a further quarter of the lengthening of the path of the real robot while navigating to a goal. Again, this is a small contribution to the overall increase of the path length.

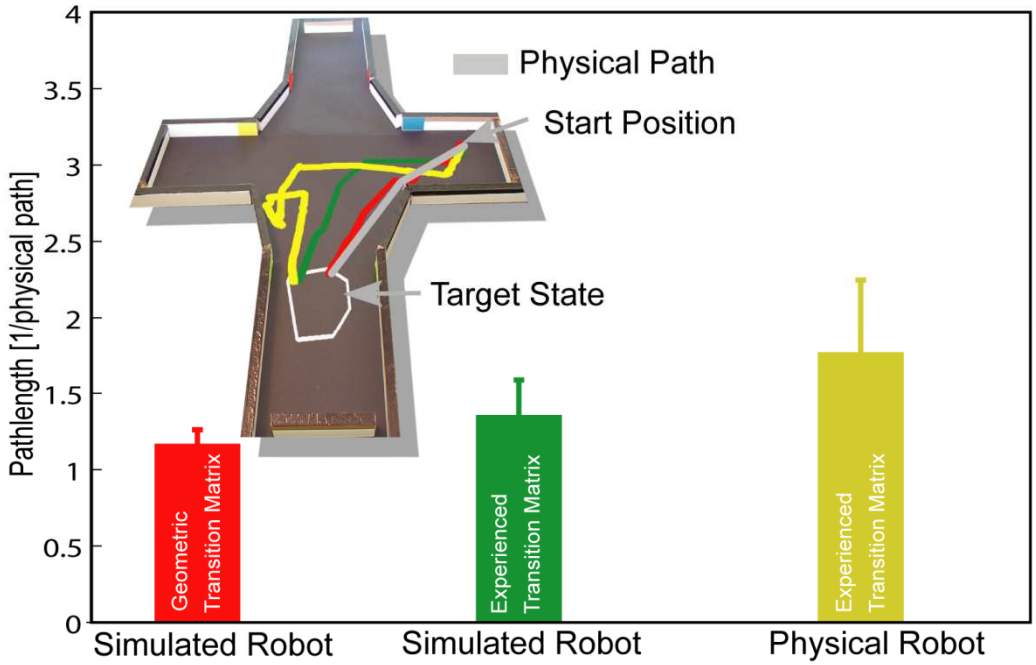

Figure 5: Navigational behavior of the robot was investigated by measuring the length of the path to different goals. The direct path, defined as the shortest traversable path from the start point to the goal state (shown as the gray line in the upper part), was used to normalize the length of the robot's path (yellow line) to the goal. The red line corresponds to the length of a path by a simulated robot by taking the topographical distribution of states (geometric transition matrix) into account. The bars represent the median length between different starting and goal states and their standard deviation.

How can we interpret the robot's navigational behavior? Approximately a quarter of the increase of the robot's path to a goal was caused by the discrete states used to represent the environment. Another quarter of the lengthening can be explained by the differences between the geometrical properties of the environment and those learned by the robot. We also analyzed the effect of obstacle avoidance on the robot's performance. The agent engaged its obstacle avoidance behavior in 60% of the trials, independent of the particular combination of start and goal states. Analyzing only the trials in which the agent did *not* engage obstacle avoidance, we obtained a median path length of 1.36 (±0.23), which is similar to the length measured in simulation with the transition matrix learned by the real-world robot. This is due to operational differences – particularly in obstacle avoidance behavior – between the robot and the simulation (see Method section). Thus the median path length given by the robot's learned transitions represents an approximation of the contribution of obstacle-free navigation. Consequently, the largest share of the lengthening of the robot's path compared to the direct path is due to the obstacle avoidance behavior, which was usually triggered when the robot moved through the narrow arms of the maze. In all configurations of goal states and start positions, the robot was able to find its goal in a reasonably short amount of time, with the main increase in path length arising from obstacle avoidance behavior.

2.2.2 Analysis of the learned transition probabilities

The robot's performance in this navigation task is a direct result of the underlying decision-making process. This process is based on the learned transition and reflex factors, which represent the learned environmental properties. Here we investigated the characteristics of the robot's learned transitions by looking at: i) the differences between the transitions of different actions on a state, ii) the influence of the used topographical distribution of states on the learned transitions of, iii) the number of different states reachable by the different actions, iv) the predictability of the state reachable by a single action execution, and v) the effect of the robot's limited learning time on the learned transition probabilities. For the most part, we analyzed the characteristics of the transition matrices by comparison to the simulation based on the geometrical transition matrix (see 2.1.2), which only takes the used topographical distribution of states into account. This comparison allows us to investigate the extent to which the topographical distribution of states gives rise to the investigated characteristics of the transition matrix.

Here we analyzed the similarity between the transitions of different actions, defined as the redundancy of the robot's possible actions on a state, by comparing the transition probabilities associated with these actions. For this purpose we computed correlation coefficients (see 2.1.2 and Figure 6A,B) between the transition probabilities of the different actions on each state. Higher correlation coefficients (>0.5) were more frequently observed in the experienced transition matrix (44%) than in the geometrical case (25%), (Figure 4A). Thus, the robot's real world action execution resulted in more similar outcomes and a higher redundancy of the actions, as compared to the geometrical case. Most (93%) of the highly correlated actions in the experienced case were obtained for states at the boundaries of the environment, and so were primarily due to the obstacle avoidance behavior elicited by wall contact. Overall, the robot's action execution resulted in more similar transitions compared to the transitions based only on the topographical distribution of states.

Next we investigated the influence of the topographical distribution of states on the robot's learned state transitions. Because the topographical properties of the states we used are fully represented by the geometrical transition matrix (see 2.1.2), we took each state and action and calculated the correlation coefficient between the transition probabilities stored in the geometrical matrix and those stored in the robot's experienced matrix. Across all actions and states, a mean correlation coefficient of 0.56 (±0.52) was obtained. Although these correlation coefficients are low, they should be considered as a conservative estimate of the similarity of action outcomes. This is because the calculation of these coefficients is based only on the transition probabilities to directly neighboring states. However, the transition probabilities to more distant states are mostly zero for all actions, and if these transition probabilities were also included in the correlation calculation, the similarity of different action outcomes would increase. In summary, while the different actions executed by the robot resulted in similar transitions more often than expected when only the topographical properties of the states are taken into account, the topographical state distribution nevertheless had an influence on the robot's learned transitions.

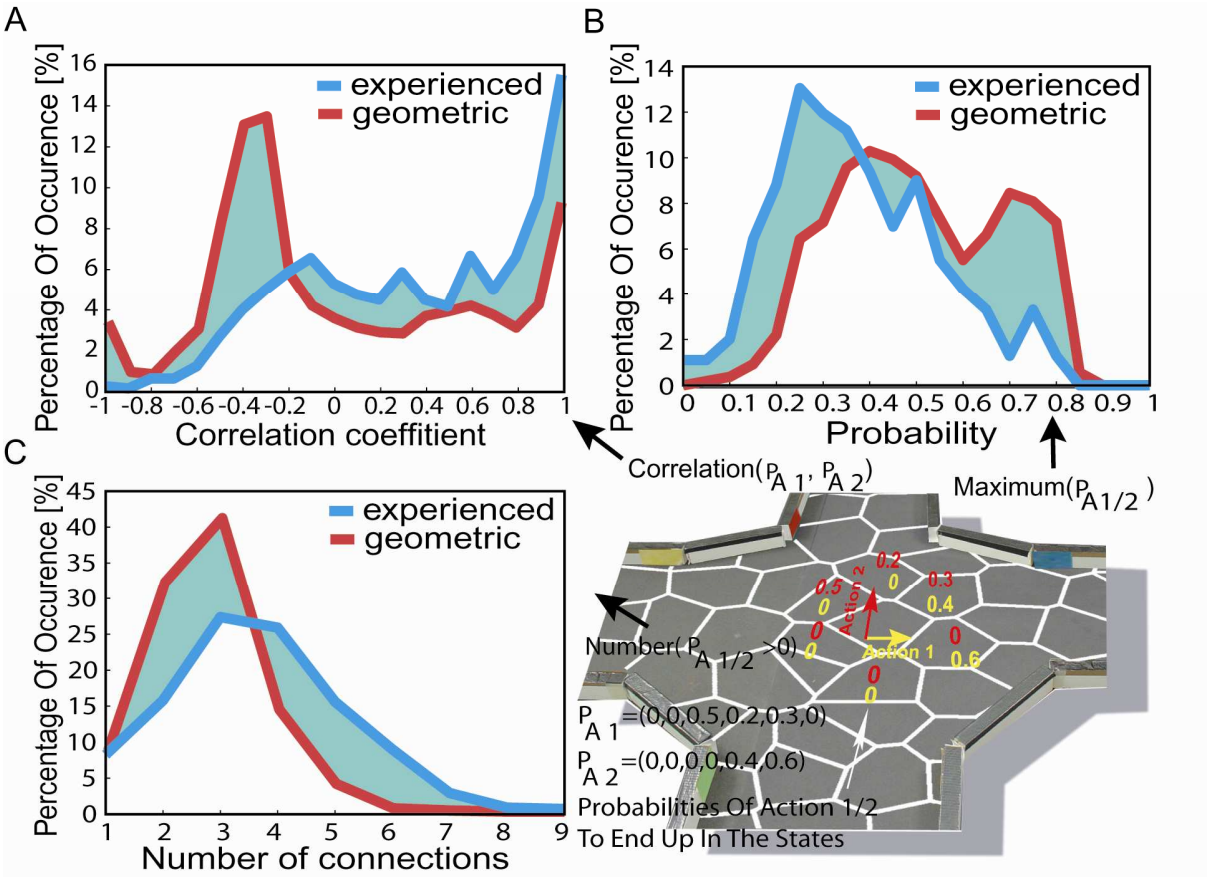

Figure 6: (A) Occurrence of correlation coefficients of the different actions. In order to do so we correlated the transition probabilities to the neighboring states of the actions as shown in the example. (B) The occurrence of the action's highest transition probability, defined as the actions predictability. (C) The actions connectivity, defined by the number of nonzero transition probabilities for each action and state.

How many different states can possibly be reached by means of a single action? To answer this question we examined the connectivity of the actions, by counting the number of non-zero transition probabilities. Figure 6C shows the occurrence of this connectivity in the geometric and experienced transition matrices. The experienced transition matrix is characterized by a higher connectivity, with more than half (51%) of all actions showing a connectivity larger than 4 in the experienced case compared to less than a fifth (19%) in the geometric case. In the experienced case the mean connectivity was higher (3.60) than in the geometrical case (2.75). The higher connectivity in the experienced case is due to the obstacle avoidance behavior – in 79% of the experienced actions which led to more than 4 connected states, the robot had to use the obstacle avoidance behavior at least once. More states can be reached by executing a single action in the experienced case.

We then analyzed the predictability of action outcomes. Predictability defines the ability to predict the state that will be reached by a given action execution, and is thus useful for action planning to perceive a certain sensory outcome. In order to evaluate the actions' predictability we introduced predictability values (see 2.1.2) proportional to the maximum transition probability of an action. Although there are alternative ways of measuring predictability (e.g. as the sparseness of transition probabilities), these are similar to the measure used here, because of the normalization of transition probabilities to one. Figure 6D shows the

occurrence of predictability values for the experienced and geometric transition matrices. Lower predictability values (<0.3) of the actions occurred more often in the experienced case (37%) compared to the geometric one (13%). Thus in general, the robot's actions are equally likely to reach a number of spatially adjacent states. This is due to the actions' transition probabilities being characterized by a non-sparse probability distribution. Furthermore we investigated the influence of the obstacle avoidance behavior on the action predictability of the experienced transition matrix. Most (84%) of the low predictability values are due to actions for which the robot had to use its obstacle avoidance at least once. In other words, obstacle avoidance reduced the predictability of the action result. In most cases we obtained a lower predictability of the robot's resultant state than we would have expected from the topographical distribution of place fields.

Next we investigated the influence of the obstacle avoidance behavior on the robot's learned transitions. The above investigations of the robot's transitions revealed a reduction in the predictability of the robot's actions, and an increase in the similarity between the robot's action outcomes when compared to the transitions based on the topographical state distribution. These effects on the transitions were due to the robot's engagement of the reflexive obstacle avoidance during these transitions. In other words, the obstacle avoidance behavior acts upon the robot's experience-gathering behavior, thwarting the actions the robot intended to do. Here we investigate the characteristics of the transitions influenced by the reflexive behavior. The obstacle avoidance behavior is guided by proximity sensors, whose activation is highly dependent on the angle of the sensors to an obstacle. These angles can change between different trials, resulting in different sensor activations and thus in different movements of the robot. Thus the outcome of the actions combined with obstacle avoidance has a low reproducibility. A direct result of this low reproducibility is that the transition probability associated with this action will be low, given a high number of experiences. In contrast, a low number of experiences can mean that the transition probabilities of these interrupted actions is high, and will thus have a high influence on the navigational behavior. In order to analyze these effects on the transition probabilities we introduced the notion of a bad connection, defined as a low correlation between the robot's intended actions and the transition that was learned, namely the action's outcome. In order to quantify this relation we calculated the line between the points within a certain state at which the robot chose an action to the point within another state, at which the subsequent action was chosen. This line was compared with the direction of the action the robot intended to take. The executed action was defined as a bad connection if the angle between the line representing the robot's traversed path and the direction of the intended action exceeded 135 degrees. We chose this threshold because the actions with this difference in orientation had a mean correlation of 0.19 (±0.51). In Figure 7A the mean transition probabilities of these bad connections in the experienced transition matrix as a function of the overall number of gathered experiences are shown. At a low amount (some hundred) of exploration steps, the average transition probability for these transitions was 0.69. With increasing exploration time, these average probabilities decayed to 0.16. The analyzed connections amounted to 29% of all connections associated with obstacle avoidance behavior. Thus, the influence on the transition matrix of obstacle avoidance

resulting in a low correlation between intended and executed action reduces with an increasing number of experiences.

Are the differences between the geometrical properties and those learned by the robot due to the robot's limited experience time? As outlined in the Method section, the geometrical transition matrix was generated by simulating the execution of each action on each position within a state. In order for the real-world robot to learn its environment to this extent simply by executing actions at random, it would have to experience the environment for an infinite time. In contrast, the robot's experienced transition matrix is based on executing each action on each state 11.54 times on average (executing actions on 1.8% over all possible positions within a state; 97% overall action execution was executed only once on a position). Here we investigated the influence of this limited experience on the robot's reduction in action predictability and the increase in the similarity between the outcomes of different executed actions. To do so, we compared the action predictability and action similarity of generated geometrical transition matrices to the geometrical transition matrix. The generated geometrical transition matrices were calculated in the same way as the geometrical transition matrix; however, the number of actions executed on each state was restricted to that of the real-world robot. We simulated 300 generated transition matrices. In order to investigate the influence of finite experience on action predictability, we calculated the action predictability values for each action of the 300 generated transition matrices. We correlated each of these 300 distributions of predictability values with that of the geometrical transition matrix and found a mean correlation of 0.89 (±0.02). In contrast, we obtained a lower similarity (r = 0.48) between the distribution of the predictability values of the robot's experienced transition matrix and the geometrical transition matrix. The same approach was used to correlate the distributions of action similarity values of the generated transition matrices with that of the geometric transition matrix, yielding a high mean correlation of 0.93 (±0.02). In contrast, a low correlation coefficient (0.42) was found between the experienced and geometrical transition matrices. Thus, restricting the amount of experience to that of the robot has a minor effect on the generated geometric transition matrices. Finally, to directly compare the geometrical and generated transition matrices we correlated the transition probabilities for each action and state of the generated matrices with the geometric one. Averaging these correlation coefficients for each generated transition matrix yielded a distribution with a mean value of 0.86 (± 0.01). A lower mean correlation coefficient (0.56) was obtained for the same correlation between the robot's experienced and the geometric transition matrix. The difference between the transition matrix constructed from the robot's experience and the geometrical transition matrix is thus dominated by the behavior of the robot and is not due to limited knowledge of its world.

Here we have investigated the properties of the transition probabilities learned by the robot. In comparison to the transition probabilities given by the topographical distribution of states, we obtained in general a lower predictability of the outcome of the robot's actions, as well as a higher similarity between the outcomes of different actions. These effects are mainly due to the real-world robot's obstacle avoidance. However, despite the differences observed between the geometrical and experienced transition matrix, an influence of the topography of

states on the robot's experiences was nonetheless observed. These properties of the transitions are due to the robot's behavior and not to the time-limited experience of the environment. Another influence of the obstacle avoidance behavior on the learned properties of the environment is given by the low correlation between the intended action and the executed action. This influence decreases as the robot increases its experience of the environment. Neglecting the reflex factors (obstacle avoidance behavior) occurring during the decision-making process, which navigational behavior would result by taking only the learned transitions into account? We would expect that it is not important for the robot to choose a precise action when moving towards a goal, due to the low action predictability as well as the high similarity between the transition probabilities of different actions. Nevertheless, the transition matrix is influenced by the geometrical distribution of the place fields, while the obstacle avoidance behavior causes a similarity between the actions and a low predictability of an action's resultant state.

2.2.3 Decision-making process

The decision-making process involves the selection of actions in order to move to a goal, and integrates the centrally learned properties – namely the transition probabilities – and the distal learned properties – namely the reflex factors. Here we investigate the impact of distal processing on the agent's decision-making process: first in terms of the frequency of obstacle avoidance behavior engaged in by the robot; and second by investigating the influence of the reflex values on the decision-making process. The frequency of obstacle avoidance behavior was quantified as the ratio of transitions combined with reflexive events to the total number of transitions. Figure 7C shows the percentages of occurrence of these reflex values for the geometric and experienced case. Higher values of these ratios occurred more often in the experienced case, with a mean value of 0.42, than in the geometrical case, reflected by a mean of 0.17. This difference in means is caused by operational differences between the robot and the simulation, such as the spatial extension of the robot (see 2.1.2), which meant that the robot-based agent used the obstacle avoidance behavior more frequently.

Next we investigated the impact of the reflex factors on the decision-making process by analyzing the normalized activity. After flooding (see 2.1.1.4), the normalized activity of a state is defined as the ratio of the maximum action activation to the sum of all actions' activations. During the decision-making process, the agent selects the action most highly activated at the robot's current location, which means that a low normalized activity describes a situation where all actions would result in a similar navigational performance. In contrast, high values define a decision-making process in which the agent chooses a precise action in order to move to the goal. In general, this normalized activity was higher for the experienced than for the geometric transition matrix (Figure 7B). This implies that the robot chose a precise action in order to move to a goal, and underwent a stable decision-making process. However, as discussed above, we actually expected a lower normalized activity considering only the transition probabilities. In contrast the lower reflex factors in the experienced case are due to an increase of normalized activities for the experienced transition matrix. Thus taking the reflexes into account reduces the effects of the obstacle avoidance behavior on the decision-making process, and results in a more precise action selection.

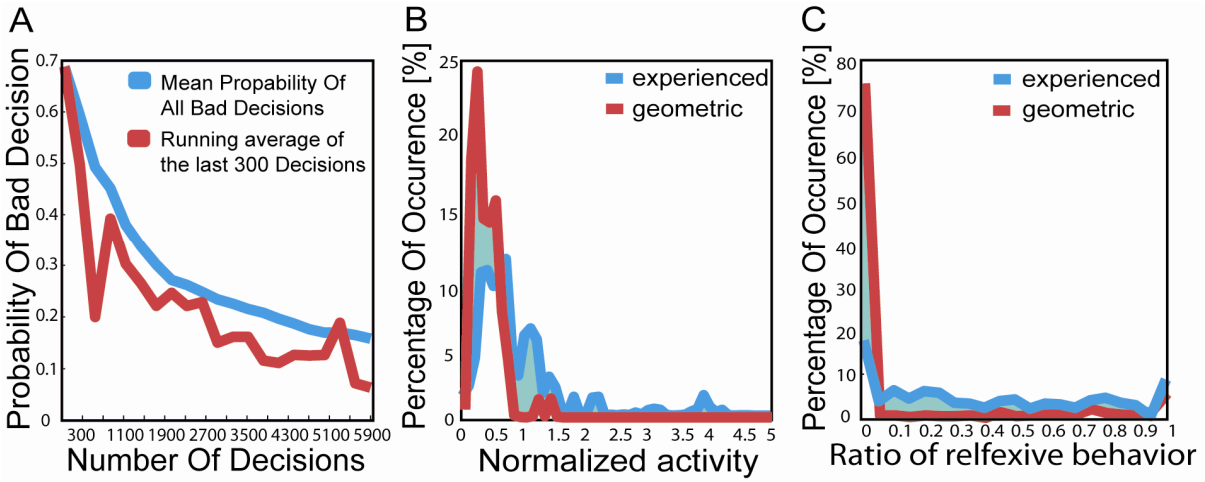

Figure 7: (A) Mean contribution to the transition probabilities of the bad decisions. Bad decision is defined as a low correlation between actions outcome and the direction of the intended action. (B) Occurrence of normalized activity in the decision making process. (C) Ratio of executed actions which resulted in reflexive behavior.

How do the different components of the algorithm influence the behavior of the robot? Taking only the central processes, namely the state transitions, for the decision-making into account, different action executions would result in similar navigational performances; although navigation in the narrow arms requires precise actions in order to reduce wall collisions and thus reduce the path length to the goal. Integrating the distal learned environmental properties, namely reflexes, into the decision-making process, the robot now executes one precise action to navigate towards the goal. Thus as we expected, taking the distal processing into account reduces the effects of reflexive behavior and allows the robot to successfully navigate in the environment. As mentioned above, another influence of the reflexive behavior on the transition matrix was a low correlation between intended and executed actions. This influence depended on the extent of the robot's experience in the environment. Taking the reflexes into account reduces the number of experiences needed to neglect this effect on the navigational behavior, as the probabilities combined with obstacle-avoidance behavior were reduced by the reflex factor. Thus the precise selection of an action in the decision-making process and the reduction of the number of experiences needed to navigate in the environment are due to the differentiation between a distal processing represented by the reflex values, and the central processing represented by the transition probabilities between the states, which are both integrated in the decision-making process. Differentiating between reflexive and central processing allows the robot to successfully navigate in the environment.

2.2.4 Learning behavior

Next we analyzed the plasticity of the navigation system by examining the adaptation of the robot's navigational behavior to changes in the environment. In order to do so we inserted an obstacle into the previously learned four-arm-maze environment, as shown in Figure 8A. We implemented and compared two different approaches to allow the robot to adapt to this change: batch and online learning (see 2.1.3).

First we investigated the adaptation process with the help of online learning by analyzing the robot's path passing the added obstacle. In each trial the robot navigated from one start state within one of the three arms to a target site, as shown in Figure 8A. In order to evaluate these trials we calculated the robot's normalized path in a certain area surrounding the wall shown in Figure 8A. Here the normalized path is given by the robot's path in a certain area surrounding the wall, normalized by the direct path, which is the shortest traversable path between the robot's entry and exit point of this area. The lengths of the robot's paths are shown in Figure 8B as a function of trial number. After a few trials (8) the path length of the robot reached values comparable to the navigational performance reported earlier. After 20 trials the selection of actions during decision-making on the different states is stabilized, and thus the changed environmental properties are fully integrated. The variation of the path length in later trials is due to the different start positions of the robot within the different arms (see Figure 8B). Thus, online learning enables the agent to quickly adapt to environmental changes.

As already mentioned, the navigational behavior of the system is based on two different environmental properties, stored as transition probabilities and reflexes. In order to analyze their contributions to the environmental adaptation, we compared the decision-making process based on transition probabilities and reflexes to that utilizing the transition probabilities alone. Thus we run the flooding algorithm (as described in 2.1.1.4) first using the transition matrix and reflexes, and second utilizing only the transition matrix. In order to analyze the integration of the changed environmental properties into the transition matrix, we compared the action selection based only on the transition matrix before learning and after all trials of online learning. As seen in Figure 9A, we found no difference but on two states between the best action of each state, selected only on the basis of the transition matrix, before (blue arrows) and after online learning (red arrows). Thus experience with the added obstacle is not fully integrated into the transition probabilities. In contrast, the action selection process based on both the transition matrix and the reflexes (yellow arrows) did show integration of the new environmental features. Thus, during online learning, the reflexes are responsible for the integration of the new obstacle into the decision making process.

We further investigated this integration of the changed environmental properties into the reflexes rather than the transition probabilities. The adaptation processes of the transition matrix and reflexes are dependent on the amount of actions already executed before the environment was changed (see 2.1.1.4). As shown previously, without taking the reflexes into account, actions share a similar activation value after the flooding process. In order for the transition matrix to adapt to environmental changes, the connectivity among neighboring states must change. However, we know that connected states share a similar activation due to the similarity and low predictability of the respective action outcomes, which means that any change in transition probability must be reasonably large in order to allow another action to be selected during the decision-making process. Before we added the obstacle to the four-arm-maze, the robot had experienced its environment by executing each action on each state 11.54 times on average. Thus, the robot would have to experience the changed environment

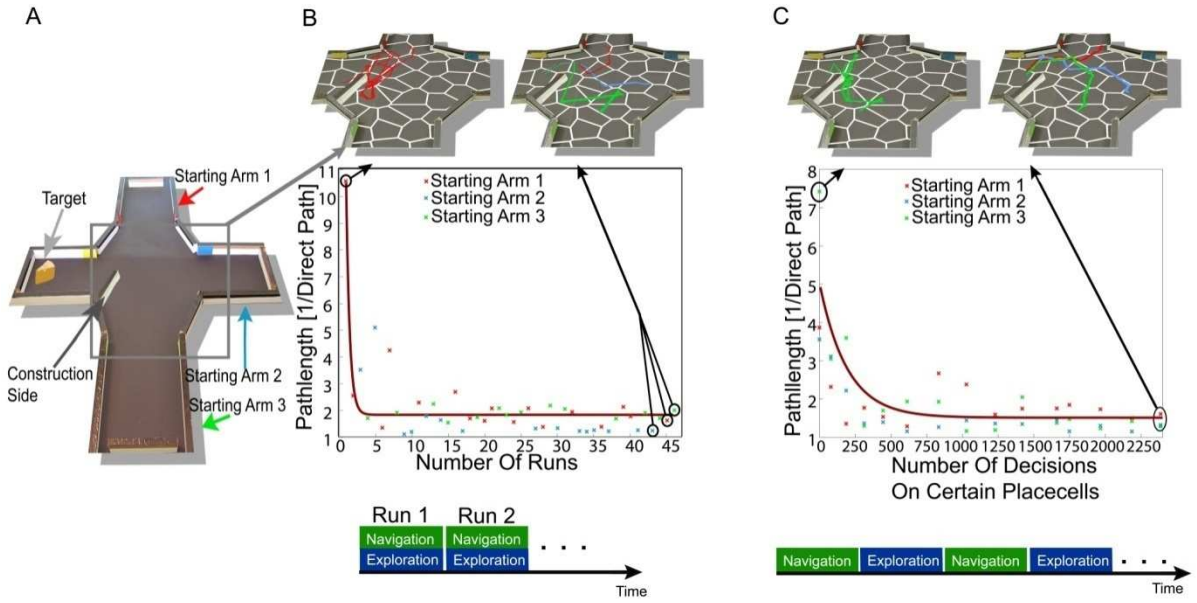

Figure 8: (A) A new obstacle (construction side) is added in an already learned environment. To test the adaptation process we evaluated the robots path through the construction side to the target side starting from three different start states, each located in one of the three different arms. (B) The adaptation process was done with online learning. This type of learning corresponds to an Exploration and Navigation stage at the same time, as shown in the lower part of the figure. The pictures in the upper part show the robot's path in an area around the added obstacle, while the robot navigated to the target. The robot's path length in this area is plotted in the center of this figure. The different colors correspond to the robot's start position in the different arms as shown in A. The brown line corresponds to the best approximation of the path length of all runs by an exponential combined with a constant. (C) The adaptation process using batch learning. After some action executions done in one of the 22 states surrounding the construction side the path of the robot from three different start states to one target side was evaluated. The robots path is shown in the upper part of the figure, for different number of experiences. In the lower part the best decision in order to move to the goal is shown for each state for a different number of experiences.

for a long time before the change in transition probabilities could trigger an alternative action selection during the decision-making process. In contrast, the reflexive processing acts as a penalty on the action's activation. Thus, the influence of the transition probabilities on the decision-making process depends directly on the activation of the neighboring states. In contrast to the influence of the reflex factors, which depends on the sum of the incoming activation for each action and thus conveniently require fewer experiences to integrate any environmental changes. As soon as the reflexes have adapted to the changes in the environment, the action which leads to a reflexive behavior is no longer executed during the online learning process. As a result, the adaptation process stops. In summary, during online learning it is the reflexive processing that enables a fast integration of the environmental changes.

Next we investigated the adaptation process involved in batch learning and compared it to online learning. Each of the experience stages was specified by the number of action execution done on each of the 22 states surrounding the added obstacle. In order to evaluate the robots navigational performance, the robot navigated in each navigation stage two times from three different start positions within one arm to the target site. The average normalized path within a certain area around the added obstacle, for each start position is shown in

Figure 8C. After an experience stage containing around 500 experiences, the path length of the robot reached a length comparable to the robot's best navigational performance reported earlier. A slight increase in the path length can be seen after 800 experiences. This is due to

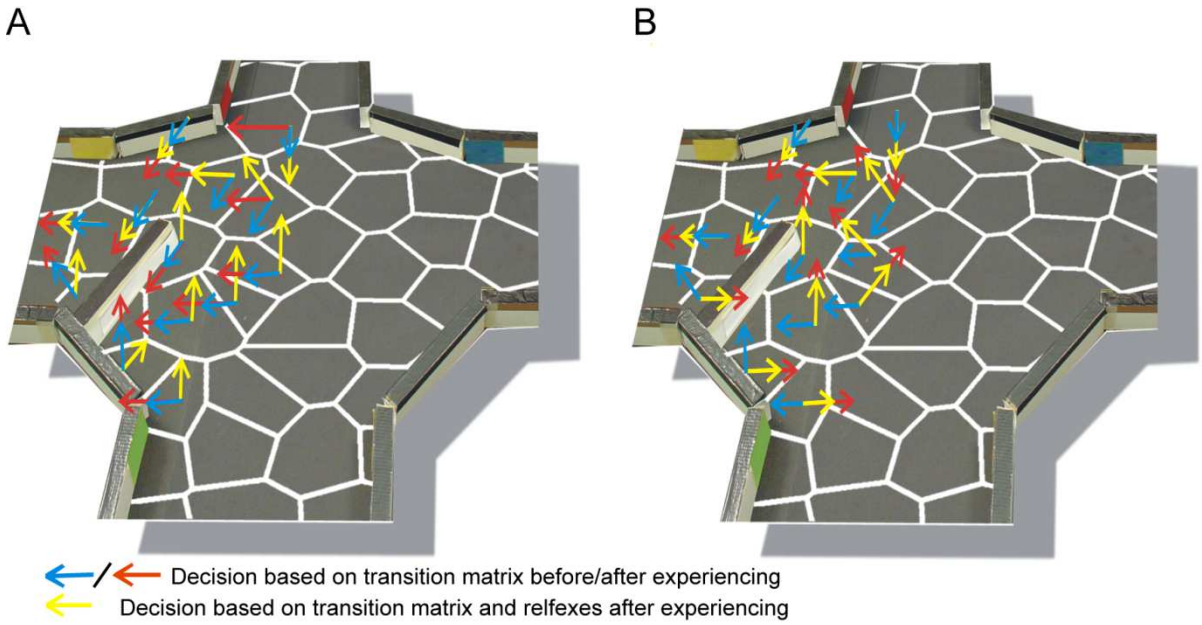

Figure 9: The best decision in order to move to the target side. The different colored arrows correspond to the different conditions. Before the robot experiences the changed environment we executed the decision making process based only on the transition matrix without the reflex factors (Blue arrow). After the adaptation process (online learning: 45 runs/ batch learning: 2500 decisions see figure 6) the decision process was calculated also without reflex factors based on the transition matrix (red arrow). The yellow arrows correspond to the condition of the integrated changes of the environment combined with the transition matrix and the reflexes. (A) represents the online learning while (B) batch learning.

some new learned features of the environment. As the obstacle is located in the middle of a state, actions are executed which results in a reflexive behavior on one side of the obstacle and on the other side not. Thus in this case batch learning could result in some instability in the decision making process, according to some conflicting experiences learned in the environment. After 1500 randomly executed actions the navigational performance does not change much anymore and in general the best actions in order move to the target did not change anymore. Thus after a short amount of time the changed environmental features are integrated in the navigational performance by batch learning. Here, the robot needed more time to experience the environment compared to online learning. This is due to the difference types of learning, as online learning integrates only the environmental features in order to move to the goal while batch learning is latent learning and thus could integrate any changed features. However after a short amount of time online and batch learning integrates the environmental changes in their navigational behavior.

Also here we analyzed the contribution of the transition probabilities and the reflex factors to the navigational adaptation. As done for online learning we analyzed the decision-making process by comparison of the decision-making process based on the transition matrix and reflexes with the one based on only the transition matrix. We concluded from figure 9B that

also the transition probabilities adapted to the changes in the environment. Thus in contrast to online learning, during batch learning the transition probabilities are able to integrate the changed environmental features.

Differentiating between reflexive and central processing allows the robot to successfully navigate in the environment. This differentiation also results in a fast integration of the environmental changes and thus navigational adaptation to the changes in the environment. The presented architecture is able to successfully model navigational behavior and keeps its plasticity in an already learned environment.

2.2.5 Four-arm-maze task

Here we modified the agent as described in section 2.1.6 in order to let the robot perform a four-arm-maze task. The four-arm-maze task is in general conducted with rats, where the rat has to find a reward (food) located at the end of one or two arms. During a training phase the rat has to learn to associate the reward with a particular label at the entrance of this arm, such that the rat only enters the arm with a reward. The rat should perform such a behavior independent of the location of the arm with the reward within the maze and thus the rat enters only the arm with the label associated with reward also after interchanging the labels at the arms. These experiments are in general done with tactile labels, such that the rat is able to distinguish the different stimuli with its whiskers. Here we modified this experiment by interchanging the tactile stimuli by visual stimuli, which the agent is able to recognize by the

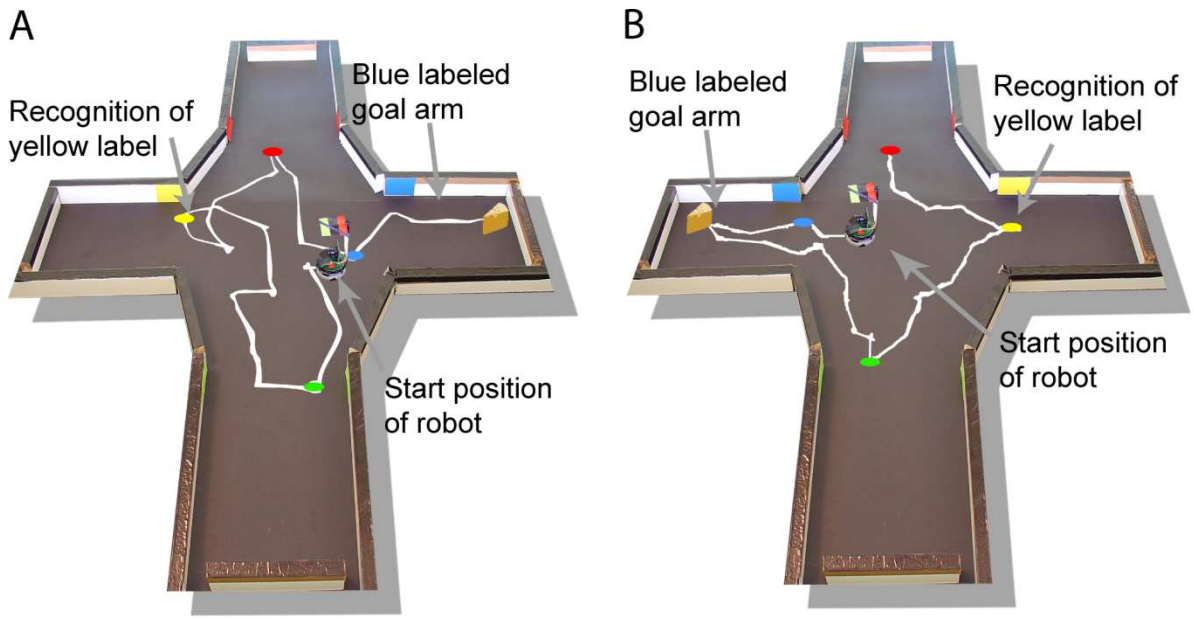

Figure 10: The four arm maze task. A: The aim of this experiment is to find the food (cheese) located at the end of one arm, here the blue labeled arm. In this experiment the robot has to enter only the blue labeled arm. Further, the agent is only able to recognize the different labels' colors in a small region around the arms' labels. Because during the different trials the labels can be interchanged or the number of blue labels can be increased, the robot has to go to detect the label of each arm to detect the label associated with food, and collect the food. The colored circle in the arena corresponds to the position the different colored labels are detected by the agent, while the white line corresponds to the robot's path during this experiments. B: Interchanging the different labels resulted in the same behavior, such that the robot only enters the blue labeled arm.

visual input of the camera attached on the robot, shown in figure 1. Similar to the tactile stimuli, these visual labels can only be recognized in a small area around the arms' labels. Based on the architecture of the agent, it is able to perform such a task, as shown in figure 10 A. This figure is based on an experiment where the blue label is associated with food and the robot only enters the blue labeled arm. The colored dots in the figure 10 defines the position, the agent recognized the arms' colored labels. Interchanging the labels of the arms resulted in the same robot's behavior, as shown in figure 10 B. Thus, with small changes of the architecture the agent is able to perform different behaviors with highlights the flexibility of the introduced behavioral model to being able to perform a whole variety of behaviors.

2.3 Discussion

2.3.1 Summary of the results

We introduced a cognitive model capable of generalizing over a broad variety of behavioral domains, and applied it to solve a navigational task. Here, navigation behavior was modeled as state transitions in a state space spanned by place cells. The architecture of this model differentiated between central processing and distal processing, where distal processing refers only to state transitions triggered by reflexive, sensory-driven obstacle avoidance, while central processing acts on all learned transitions between states. Reflexive behavior shields the robot's learned transitions from the complexities of the environment, resulting in uniformly distributed and hence less predictable action outcomes than expected from inspection of the topographical distribution of place fields used. Integrating the information gained by reflexive and central processing in the decision-making process reduced the impact of sensory-driven obstacle-avoidance behavior on the navigational performance. In addition, the introduced model was quickly able to adapt to changes in the environment. Consequently, the robot was able to successfully navigate in the environment after only a short amount of time.

2.3.2 Generality of the behavioral model

The introduced cognitive model is based on a low-dimensional sensory representation composed of discrete sensory states. In this state space the robot first had to learn the sensory outcomes of its action's execution, namely the state transition and the reflex factors. Thus, the robot learned the environmental properties with respect to its actions in an unsupervised fashion. Based on the acquired knowledge, the robot planned its action in order to navigate to the goal state. Sensory states were defined equivalent to place fields of place cells found in the hippocampus, providing a representation of body position within the external space. From a signal processing point of view, place cells can be understood as an optimally stable representation of visual input from a robot moving in an environment (Wyss et al., 2006). That is, unsupervised learning can reorganize the sensory space spanned by the robot's visual input, leading to a low-dimensional representation of the sensory input with a spatial interpretation. We determined the robot's current state from the activation patterns of multiple place fields using a winner-takes-all process over all place cells' current activations. Thus we obtain a discrete division of the robot's navigational space, now spanned in terms of the robot's possible positions. Since this navigational space completely determines the navigational task, a discrete division of this space corresponds to a discrete division of any

sensory space relevant for navigation. A further sensory signal and sensory space used here is spanned by proximity sensors measuring distances to objects. Proximity sensors allow differentiating between distal processes – which sensory transitions were influenced by the sensory-driven behavior – and central processes – the state transitions. Our model can be extended to other types of behavior by division of the relevant sensory space into discrete states and implementation of the actions that define the behavior[3]. Additionally, here we defined a number of discrete actions such that the agent's action repertoire can be easily expanded, e.g. by lifting object by a robotic arm. By means of executing all possible actions, the sensory outcome in the state space can be learned and based on these learned knowledge a whole variety of behavior can be modeled (Wolpert and Ghahramani, 2000).

2.3.3 Comparison to biological behavior

In this navigational paradigm we used 8 different discrete actions defined as a movement in a certain direction. In contrast, animals have a much richer repertoire of actions to move in an environment. Given a state distribution as shown in Figure 1, the sensory outcome of different actions resulted in redundancies, which would be further increased when increasing the number of actions such that based on the state space the robot would not gain more information about the environment by more action possibilities. But by an increase of the action possibilities, the agent's navigational performance could be increased. Although an increase of the number of actions would result in a higher similarity between actions, the slight changes of transition probabilities could result in a more precise navigation to the target and thus a smaller path of the agent to this target. Using discrete actions allows the cognitive architecture to be easily expandable. Without a change of concepts it might be applied to a robotic arm, e.g. lifting objects. Admittedly, in a very high dimensional state space new problems due to very sparse data would arise, such as an increase of the agent's experience time to learn the transition probabilities. In this cognitive architecture the primary limitation is the robot's experience time, which increases with the number of states and actions and thus with the complexity of the behavior to be modeled.

Do agents, behaving in the real world, show a restriction of action possibilities? In our behavioral model the restriction is given by the number of action possibilities as well as by discrete actions. Different studies suggested the existence of motor primitives (Flash and Hochner, 2005) which can be sequentially or simultaneously combined to realize a certain behavior. Here, we only allowed the predefined action possibilities to be combined sequentially. In order to allow a simultaneous combination of actions, our model has to be expanded such that the agent can explore the sensory outcome for each linear combination of simultaneous actions. Thus as discussed previously, such a restriction of action possibilities limited the computational efforts of such an experience process. Further, motor learning in humans have a so called problem of degree of freedom (Bernstein, 1962), caused by the number of muscles and joints of the human body, thus the huge amount of different possibilities to perform the same action. Bernstein suggested by maintaining some joints from moving, in a first step of motor learning a reduction of the degree of freedom is taken place.

[3] In section 3 of the thesis we suggested an unsupervised division of the sensory space into discrete states by increasing the predictability of the sensory outcome of the agent's action.

This restriction of the degree of freedom was also validated in the motor learning in humans (Vereijken et al., 1992). Thus, in motor learning, restriction of the action possibilities can also be found in biological agents.

In this study we applied our model to navigational behavior. The model is not able to perform all varieties of navigational behavior, but can only perform navigation in previously explored environmental regions. In contrast, animals are also able to perform cue based navigational tasks, i.e. to navigate to a visual cue without previously exploring the particular environmental regions which the animal has to pass on his way to the target. To do so, animals must transfer learned knowledge from similar environments to the current one. Although these processes have not been fully understood, it can be hypothesized that sensory information independent of the environment, like vestibular sensory input, is necessary to accomplish this task. In the presented navigational model we did not use any information independent of the environment and thus do not allow cue based navigation. Here we hypothesized that a change of sensory input of the model could result in different types of navigational behavior, like cue based navigation.

Here we applied a probabilistic approach the agent's sensory outcome of its actions, by transition probabilities. Previously, the concept of transition probabilities is applied in reinforcement learning (Sutton and Barto, 1998), and in recent years this concept was employed in neuroscience (Wolpert et al., 2000) for biologically inspired behavior. Noise in the motor system converts into noise in the sensory outcome of actions, and thus sensory consequences of actions should be described by a probabilistic framework. Here, the learned knowledge about the environment was stored in state-to-state transition probabilities, on which our decision making process is based on. This decision making process is modeled by executing locally on each sensory state the action with the highest probability to reach the target. Thus, the maximization of the optimality criteria, defined by the highest probability to reach the target, resulted in a choice of a particular action. Such an optimality criteria was also found in biological systems (Todorov, 2004). Further, the choice of the sensory state had an influence on the values of the transition probabilities. The sensory states were low dimensional sensory representation, which can be understood as resulting from a reorganization of the sensory space, as seen in the following section 3. Furthermore, with the help of this probabilistic approach we were able to successfully model a target related behavior like navigation. This probabilistic approach could be generalized to capture a variety of natural, goal-directed behaviors.

2.3.4 Comparison to computational models of behavior

Studies that have modeled navigational behavior on the basis of place cells as a representation of the environment can be characterized by the type of learning used: Hebbian learning or reinforcement learning. Hebbian learning exploits the fact that moving in an environment activates more than one place cell at the rodent's location, as place fields of the corresponding cells overlap. This allows the application of the biologically motivated principles of LTP and LTD to strengthen connections between place cells active in a certain time interval. Connected place cells represent a cognitive map (Gerstner and Abott, 1997; Blum and Abott, 1996; Gaussier et al., 2002). Other studies introduced a cell type - goal cells

- representing the goal of the navigational task (Burgess et al., 1997; Truellier and Meyer, 2000). The connections between a place cell and the goal cell encode the place cell's direction to the goal, also derived by the modulation of connections between these two cell types through Hebbian learning. In contrast to our model, the mentioned modeling approaches rely on global orientation and a metric that allows measuring the direction and distance to the goal from a given location within the environment. The global orientation used by these studies is gained by using the same frame of reference over the whole environment. In contrast, we aimed for an approach with fewer assumptions that does not require the introduction of global variables such as orientation or a metric. Instead the robot is able to learn the topology of the environment through experience. Similarly, some of the mentioned studies (Strösslin et al., 2005; Foster et al, 2000; Gerstner and Abbott, 1997; Burgess et al., 1997; Truellier and Meyer, 2000) used population coding to encode the position or direction to the goal. However, the population vector approach is based on the assumption that place fields and rodent's orientations have separate topologies. Thus to decode the robot's position or orientation, the weighted average of place cells or orientations has to be calculated. This again incorporates knowledge of the topology in the decoding scheme and impedes a generalization to other action repertoires. In contrast, we defined the actions independently of each other so that the action repertoire can easily be expanded, for example including the action of lifting an object. Other branches of studies (Forster et al., 2000; Aleo and Gerstner, 2000; Strösslin et al., 2005) used reinforcement learning (Sutton and Barto, 1997) to perform a navigational task. In the concepts of Markov Decision Processes and value iteration (Sutton and Barto, 1997) are commonalities between reinforcement learning and our approach, yet in our model, value iteration was expanded by reflexes. A pure reinforcement learning approach involves learning the properties of the environment by using an explicit reinforcement signal, given by a goal state; in the presented model these properties are latently learned (Tolman, 1948), resulting in a global strategy for navigation in this environment. Hence, in contrast to other studies, here we present a cognitive model that is able to acquire the topology and properties of the environment empirically in a latent manner and can additionally be expanded to model other behaviors by redefining the meaning of the actions and states.

Also artificial intelligence (AI) tries to capture autonomous behavior as shown by animals. A comparison of our work with AI seems appropriate because the algorithmic implementation of the introduced behavioral model shares some similarity with graph theory, often used in AI. Here we roughly situate our research in this scientific field, but do however not intend to give a profound philosophical or technical discussion of AI. In the literature, a common division of AI approaches is that between old and new AI (Verschure and Althaus, 2003) where old AI is associated with processing of physical symbolic systems (Newell, 1980), supposedly relating to the explicit modeling of higher cognitive tasks such as problem solving and planning (Newell, 1990). Symbols represent a physical pattern, or as Honavar and Uhr (Honavar and Uhr, 1994) state it, symbols are condensed and discrete semantic representatives for certain piece of knowledge. These symbols are combined into structures (expressions) and manipulated to produce new expressions with syntactical operations. One of the main critics is expressed in the so called symbol grounding problem (Harnad, 1990;

Searle, 1982), asking how symbols developed their meaning. Approaches evolved which emphasize the importance of situatedness and grounding through the use of real-world systems (Brooks, 1991). Although the problem of symbol grounding is not sufficiently solved, the systems addressing this problem mostly rely on sub-symbolic approaches (Chalmers, 1992). The sub-symbolic level is commonly thought of as a level below symbolic representation, also involving the neuronal level, arguing that sub-symbolic processes will turn out to be sufficient to evolve symbol-like semantic properties without preprogramming. Thus here we roughly differentiate the field of AI into symbolic and sub-symbolic processing.

How can we classify the concepts of our behavioral model within AI? In our model the agent generates the topography of its sensory space by interacting with its environment. The dimensionality of the sensory space is reduced by low dimensional representations of place cells, resulting from the systematic characteristics of the sensory input from which invariances can be extracted by unsupervised learning (Wyss et al., 2006). It is important to note that even though from the outside these place cells map to a space in the world, for the organism place cells do not have an explicit symbolic spatial representation. Rather, they merely represent invariance within neural activation; they entail a spatial meaning only through the organism's situatedness. This becomes obvious by a small thought experiment: If an experimenter would consistently lift the robot when it is located on place field x and put it manually to place field y whenever the robot executes action 1. Let place field x and place field y be far away from each other. Because of the continuity generated by the experimenter, the agent will learn the transition probabilities between these two place fields and assume a neighboring relation between this corresponding place fields, although these place fields are far apart from each other. Hence, place cells represent a low dimensional sensory representation, but without a predisposed topographical interpretation - instead this topographical interpretations has to be learned by the agent from the contingencies in the environment. Further, the environmental knowledge learned by the introduced behavioral is specific for individual environments and does not facilitate abstract symbols or syntax which would allow deductive reasoning about future environments. Unlike to our concern, achieving generalizability through deductive processes is the main motivation for an explicit symbolic representation of knowledge in AI. Our main concern, however, is the reproduction of navigation behavior that does not require the explicit construction of generalizable symbols, but on the concrete subsymbolic knowledge an agent can learn from its sensory signals about a concrete situation. Nevertheless, in contrast to the majority of studies in the sub-symbolic domain of AI, which model a particular behavior of a particular species like flies, lobster or crickets (For a review see Webb 2001), here we introduced a general sub-symbolic model which can be applied to various behavioral domains.

In summary, the introduced model of behavior shares some similarities with different concepts of AI and reinforcement learning. But the main differences are that the agent is able to learn the topology of the sensory space according to its interaction with its environment. Further, the agent latently learns the environmental properties.

2.3.5 Why using robots?

In recent years robots became a useful tool to answer neuroscientific questions. The behavioral study of autonomous robots shares some analogies to the study of animal behavior. Both scientific fields involve agents interacting in natural environments, involving sensory processing, decision making and action execution. This is emphasized by the usage of robots in neuroscientific fields to test hypothesis, characterize problems and to investigate the environment of biological systems (Webb, 2000). Examples are robotic tests of hypothesis about the mechanism of learning (Saksida et al., 1997; Verschure and Voegtlin, 1998), on the contribution of different neurons to particular behaviors (Franceschini et al., 1992; Blanchard et al., 1999) or, on a more conceptual level, the involvement of the motor system in decision making (Wyss et al., 2004). On the other hand, robots are used to investigate the characteristics of the sensory data perceived by an animal (Kuwana et al, 1995; Wyss et al., 2006; Franzius et al., 2007) and to describe the dynamics of the bodies' interaction with its environment (Raibert, 1986). In summary, robot implementations provide a critical test of neuroscientific concepts (for a review see Webb 2000, 2001; Voegtlin and Verschure 1999).

Yet transferring the results from robot experiments to the corresponding biological systems does not come without difficulties. Although the investigation of animal's behavior and robot's behavior shares some similarities, there also exist a lot of differences. For one, the exact imitation of complex animal behavior by robots is still out of reach due to limitations in computational power and limited mechanical similarities with respect to skeleton and muscles. The complexity of behavioral and mechanical properties of biological organisms requires various simplifying assumption of the model of behavior implemented on a robot. Thus only selected aspects of animal behavior can be investigated in a given, which restricts the transfer of results to biological systems. Hence, most scientific work utilized robots to investigate aspects of the biological organism, like phonotaxis (Webb, 2000), and thus a model containing the whole complexity of the biological organism is not required. In summary, various selective properties of biological systems could been successfully investigated using robots.

Robot experiments are sometimes preformed a simulated environment. Simulations have some advantages, mostly that the robots behavior, environment and its mechanical properties can be easily changed. This allows comparatively easy investigation of the influence of these parameters on the biological system. Thus simulation is a useful tool to answer questions of Neurocybernetics. However, it is much harder or even impossible to match the whole complexity of the real world in the physical properties of simulated environment. Given that biological systems develop to cope with such noise, simplified simulated environment face their own restrictions with respect to biological plausibility. Similarly, also timing plays an important role in biological systems, and thus real time experiments in natural environments are required to understand the biological systems (Webb, 2000). However, simulations are a powerful tool to answer different biological question, a validation in the real world has to be taken into account, as the biological system is embedded in this real world and thus the model should be verified in the real world.

Here we investigated navigational behavior on a robot in a real world based on a probabilistic method to learn the sensory outcomes of the robots actions in the presence of natural noise. Thus the sensory space plays a crucial role in our model of behavior. In the model we differentiated between central and a distal processing. The distal processing component is composed of reflexive behavioral abilities for obstacle avoidance, fully controlled by sensory input. That is, proximity sensors which measure the reflection of infrared light at certain boundaries governs the behavior, yielding a direct influence of noise generated by these sensors on behavior vial distal reflexes. In contrast, most of the simulation cannot capture the complexity of this sensory input, especially the noise level, because in general simulations model this noise levels in a more structured way (for example by assuming Gaussian noise). As seen in the previous section the difference between the central component and distal component, which is due to the less structured transition caused by the reflexive behavior compared to the transition without a reflexive behavior, is crucial for the differences caused by a division between these components. Thus, we would expect that the difference between physical and simulated implementation would influence these results. In this way the sensory input given by a real world environment was crucial for this investigation and thus represents also an example of the importance to implement the behavior in a real world environment.

2.3.6 Summary

We have introduced a cognitive architecture for the modeling of animal-like behavior, and plan to use this model to obtain further insights into the general principles of behavior, such as action planning. The model is based on a discrete division of sensory space into states, here obtained by randomly distributing place cells in an environment. The topographical distribution of these states was shown to influence the state-to-state transition probabilities, thus affecting the robot’s behavior. This raises the important question of how these states should be organized to allow the robot to behave optimally. In other words, how can we optimally reorganize the sensory space in the light of the sensory input and the robot's navigational action repertoire? Recently, similar questions on the reorganization of sensory space are intensely studied in the somatosensory system, Oudeyer et al., 2007; Kaplan and Oudeyer, 2007). As our model can generalize over different behavioral domains, we claim that it can be employed to gain insights into different principles of behavioral.

3. Reorganization of the sensorimotor space

3.1 Method

Here we introduce the optimization algorithm for the reorganization of a simulated agent's sensory space. The goal of this algorithm is to increase the predictability of the sensory outcome of that agent's actions. The algorithm reaches this goal by dividing sensory space into disjoint discrete states. We will refer to these states as Macrostates and to a specific sensory space discretisation as a Macrostate configuration. The dynamics of a sensory state transition resulting from an action are captured in the form of Macrostate transition probabilities. The predictability of a Macrostate configuration is determined by examining the transition probabilities between its constituent Macrostates.

3.1.1 Simulating agent behavior

Here we applied the model of behavior developed in section 2. We modified this behavioral model by introducing different parameters to change the agent's motor capabilities and thus investigated how this motor parameters effected the optimization algorithm.

We simulated an agent moving in a number of 2D maze environments. It could execute eight action primitives corresponding to traveling a fixed distance (parameterized by *step-length*) in one of eight directions (0° to 315°). Step-length and movement direction were subjected to additive Gaussian noise (parameters: *step-length noise σ*, *angular noise σ*). The agent's sensory space was spanned by its position in the 2D environment.

The action parameters were varied to explore their influence on the outcome of the optimization process. The statistics presented in the section 3.2 are based on five optimization runs for every combination of step-length (5, 10, 15 and 20 microstates (see Figure 10)), step-length noise σ (1/3, 2/3 and 1 microstate) and angular noise σ (2°, 6° and 12.3°). There were three maze environments of comparable size (bounding box of ca. 80x80 microstates): circular, square-shaped with straight walls and roughly square-shaped with irregular walls.

3.1.2 Macrostate transition probabilities

We assumed that the agent possesses the means to acquire knowledge of Macrostate transition probabilities (Weiller et al., 2007). It could do so by simply running a random exploratory motor program if these probabilities required updating as a result of Macrostate modification. As it is computationally not efficient to re-simulate this exploration procedure after each iteration of the optimization algorithm, we made use of a short-cut by introducing *microstates*. Microstates are the atomic elements of a very fine grained discretization of sensory space as shown in Figure 11 A/B. The transition probabilities in sensory space were thoroughly sampled at this highly resolved level. As Macrostates can be seen as disjoint sets of microstates, transition probabilities at the Macrostate level can be computed from microstate transition probabilities.

For the mapping of the Microstate transition probabilities to the Macrostate transition probabilities of a configuration of Macrostate, a one step flooding process was used. The transition probabilities for the different states and actions are similar to a directed graph. Here

the states are the nodes and the transition probabilities are the weights. The transition probabilities of one Macrostate were calculated by setting the activation of the Macrostate's Microstates to one. This activation was transferred to the connected Microstates by multiplying this activation with the weights of the microstates. This propagation of the activation was done only once and this process was defined as one step flooding, and it was assumed to be similar to the agent's execution of an action from this Macrostate. Next we calculated the activity of all Macrostates, by summation of the activity of the dedicated Microstates. The Macrostate transition probabilities were calculated by normalizing the Macrostates activity by the sum of all Macrostates activities. The probability to make a transition to the same Macrostate is defined *selfconnectivity*. This process has to be executed for all action and states in order to calculate the Macrostate transition probabilities for the current Macrostate distribution.

3.1.3 Optimization process

Following the computation of Macrostate transition probabilities, these were evaluated with respect to their predictability and the degree of their decorrelation. This enabled the optimization algorithm to modify the Macrostate configuration in such a way as to improve these measures. We formalize the predictability ($pred_i$) of a Macrostate i as the sparseness of its transition probability distributions, averaged across actions. As proposed in (Karvanen and Cichocki, 2003), the sparseness of such discrete distributions was measured by their Euclidean norm.

The decorrelation of two Macrostates, decorr(i,j), was specified to be the converse (1-corr(i,j)) of the uncentered correlation coefficient of the corresponding pair of transition probability distributions, averaged across actions.

A maximally predictable, yet trivial Macrostate configuration would consist of a single state with the probability of remaining in this state being 1. In order to prevent the algorithm from arriving at this trivial solution, self-connection strength (sc_i) is to be kept low for all states. To obtain an overall measure of Macrostate quality, we introduced the objective function, Ψ. The Ψ value of a Macrostate i is a weighted sum of its predictability and minimum decorrelation ($decorr_i=min_j(decorr(i,j))$) scaled by the inverse of its self-connectivity strength. After an optimization run has been concluded, the Macrostate configuration with the highest average Ψ is chosen to be that run's result.

$$\Psi = \langle (2 * pred_i + 0.5 * decorr_i) * (1 - sc_i \rangle_{Macrostates}$$

The optimization algorithm had two procedures to change the configuration of Macrostates: Cut and Merge (shown in Figure 11 D/E). The cut procedure partitions the Microstate of one Macrostate in two subsets of Microstate, yielding two new Macrostates; each of the two new subsets has similar transition probabilities. The Merge process combines the Microstates of two Macrostate to one Macrostate.

3.1.3.1 Cut procedure

In order to split the Microstates of a Macrostate into two subsets of Microstates (Figure 11E) with similar microstates transition probability we introduced a similarity measure. We first

calculated to which Macrostates the Microstates of the one to be cut is connected to. For each of the Microstates dedicated to the Macrostate to be cut, we summed over the Microstates transition probabilities, which have a transition to each of the other Macrostate. This resulted in a measure of the projection for each of the Microstates to which Macrostates are they connected and with which weight. To investigate the similarity between the Microstate transition probabilities, we correlated this measure of the projection, resulting in a correlation matrix. Here we averaged this correlation matrix over all possible actions. Based on this correlation matrix we used the normalized cut Algorithm (Shi and Malik 2000), which groups the Microstates into two subsets. This algorithm groups the Microstate such that within a group there is a high while between the groups is a low correlation. The normalized but represents an optimal solution of these two constrains. In summary with the help of a similarity measure the Microstates of the Macrostate to be cut were grouped into two subsets, characterized by similar transition probabilities within the groups.

3.1.3.2 Merge procedure

The merge procedure combines two Macrostates to one Macrostate. In order to do so we dedicated the Microstates of the two Macrostate to one new Macrostate, as shown in Figure 10D.

3.1.3.3 Heuristic of the optimization process

Computing the derivative of the objective function with respect to changes in the Macrostate configuration is not feasible due to the high dimensionality of configuration space. As a consequence, the optimization algorithm is not driven by the gradient of Ψ, but by a number of rule-based heuristics: If a Macrostate has low predictability, it is split in two. If a pair of Macrostates is highly correlated, they are merged. These two antagonistic operations can be understood as an iterative local refinement or coarsening of the sensory space discretization aimed at producing state configurations with high average Ψ.

In each iteration, the optimization algorithm had to decide whether to cut or to merge as well as which Macrostates to subject to the selected treatment. A number of simple rule-based heuristics were employed to make these decisions. The first thing following Macrostate evaluation was to compile lists of possible *merge-candidates* and *cut-candidates*. The list of merge-candidates initially contained all Macrostate pairs. This list was refined by resolving intersections between Macrostate pairs by removing the Macrostate pair with the higher decorrelation value. Also candidate pairs were discarded that, if merged, would create a Macrostate whose self-connection strength was its largest transition probability. The cut-candidate list contained all Macrostates characterized by a predictability smaller than the median of predictability values of the current Macrostates configuration. Once the candidate lists were composed, the algorithm chose to carry out the merge operation if the mean decorrelation was lower than the mean predictability, as shown in Figure 10E/D. If mean predictability was lower than mean decorrelation, the cut operation was executed instead.

In addition to these rules, we introduced an upper boundary for the possible number of Macrostates to prevent the algorithm from reaching the trivial Macrostate configuration where each Macrostate consists of a single microstate. This boundary was periodically

changed every 20 iterations from 1000 Macrostates to 500 Macrostates to 10 Macrostates to prevent the algorithm from oscillating. Each optimization trial was initialized by randomly distribution of the Macrostates. After some thousand iterations, the optimization process was concluded.

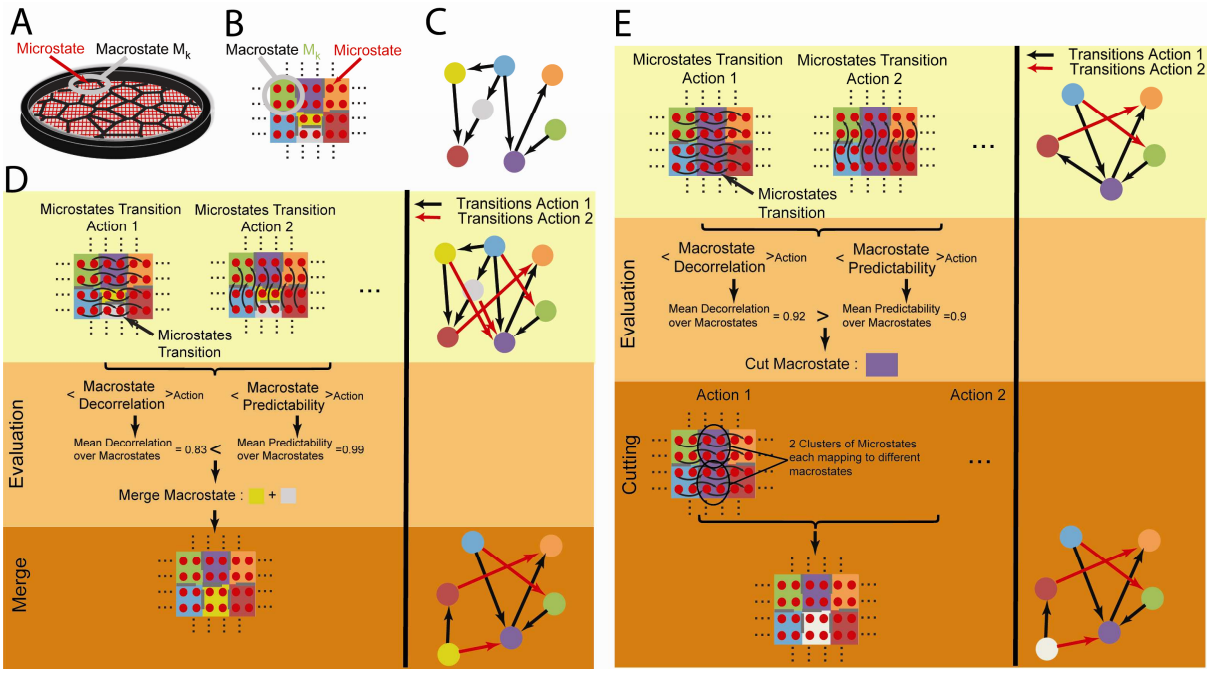

Figure 11: Heuristic rule-based procedure of the optimization Algorithm. A: Microstates are a finer discretisation of the sensory space, which is here spanned by the position of the robot. The Microstates which are surrounded by a black line represents a Macrostate. B. Each red circle corresponds to a Microstate and is assigned to one Macrostate, which is defined by one color region. C: The Macrostate and the Macrostate transition probabilities are similar to a directed graph. Each Macrostate is represented as a colored node and the transition probabilities are similar to edges, which are shown as arrows. This optimization algorithm utilizes techniques from graph theories in order to be easily transferable to other modalities. D: The left side represents the Microstate level while the right the directed graph of the Macrostates. Based on the Macrostate transition probabilities the mean correlation and predictability of the Merge- and Cutcandidates were evaluated. In case the decorrelation is lower than the predictability the two Mergecandidates (yellow and were) were merged by assigning the Microstates of these two Macrostate to one Macrostate. D: In case of a lower predictability than decorrelation, the Cutcandidate (purple Macrostate) has to be cut. The cut processed analyzed the microstate transition probabilities in order to group the Microstates of the Cutcandidate such that each group had similar transitions to Macrostates.

3.1.4 Analysis methods

In our case, sensory space is equivalent to the two-dimensional environment the agent behaves in. Therefore, each Macrostate occupies a distinct spatial region within that environment. To investigate the spatial structure of these regions, we applied three region analysis measures (Jähne, 2002). The *area* of a Macrostate is defined by the number of microstates in it. *Solidity* is a measure of Macrostate compactness and computed by dividing the area of that Macrostate by the area of its convex hull. The *eccentricity* of a region is defined by the ratio of the distance between the foci of a fitted ellipse and the length of its major axis. *Roundness* is an inverse measure of Macrostate eccentricity (1-eccentricity).

To directly compare the topographical distribution of two Macrostate configurations, we applied a measure of spatial similarity. First, we constructed a spatial occupancy map for each Macrostate. Then, each occupancy map from the first Macrostate configuration was compared to each map of the second configuration by computing the uncentered correlation coefficient. To get from this one-to-many comparison to an injective one-to-one comparison, each Macrostate in the smaller configuration was associated with the Macrostate in the other configuration which was spatially most correlated to it. Consequently, there remained a single correlation value for each Macrostate in the smaller configuration. These values were averaged to yield the final configuration similarity measure.

3.2 Results

We analyzed the algorithm with respect to four questions. Does it successfully increase the objective function? What is the spatial topography optimized Macrostate configurations? Does the hypothesized adaptation to motor parameters come about? How does optimization of predictability relate to optimization of stability? In order to investigate these questions, we applied the algorithm to the sensory spaces of a number of agents with varying motor parameters and situated in different environments (See 3.1.1).

3.2.1 Objective function

First, we will examine how the objective function Ψ and its components, degree of decorrelation and predictability, evolve during the course of optimization

Ψ is a function of the predictability, degree of decorrelation and self-connection strength of a Macrostate configuration. As the optimization process does not directly optimize Ψ using gradient-based techniques, the optimal Macrostate configuration (highest average Ψ) need not necessarily occur at the end of the optimization process, as shown in Figure 12. This Figure shows an exemplary curve progression of the objective function and its components as well as the current number of Macrostates of an optimization run. The maximum of Ψ during this optimization run is give by 1.18. In the first 100 iterations the curve progression of the objective function is dominated by the high degree of decorrelation, which is caused by the initial random distribution of Macrostates. The increase of the degree of decorrelation in the first 30 iterations steps is caused by an increasing number of Macrostates. This is due to the fact that the highest degree of decorrelation is obtained if each Microstate corresponds to one Macrostate. In contrast, during the first 100 iterations the predictability values are very small. After 100 iterations the rising predictability values have a higher impact of the curve progression of Ψ. After a couple of hundred iterations the curve progressions of the optimization measures are characterized by oscillations. These oscillations are characterized by different periods but are mainly caused by a cut and the following merge process resulting in the same distribution of Macrostate as before the cut process. These oscillations are disrupted by changing the upper boundaries of maximal possible Macrostates. The corresponding distributions of Macrostates to the different Ψ values are shown in Figure 13. In general, the curve progression of Ψ, the degree of decorrelation and predictability changes discontinuously.

Across all optimization runs, the random initial Macrostate configuration at the beginning of the run had a mean Ψ and standard deviation of 0.29 (± 0.08), mean predictability of 0.01 (± 0.005) and mean degree of decorrelation of 0.45 (± 0.14). The corresponding values across all optimized Macrostate configurations were Ψ 1.23 (± 0.13), predictability 0.516 (± 0.069) and degree of decorrelation 0.48 (± 0.08). Note that the dynamic range of degree of decorrelation is larger than it appears to be given the difference between initial (random) and optimal Macrostate configuration. As a random Macrostate configuration can be quite decorrelated, there are iterations during which the degree of decorrelation is lower than the initial value. Note also that self-connection strength stays very small (below 0.05) throughout the optimization process. It becomes clear that the optimization process successfully increases the objective function Ψ as well as its major components, most notably predictability.

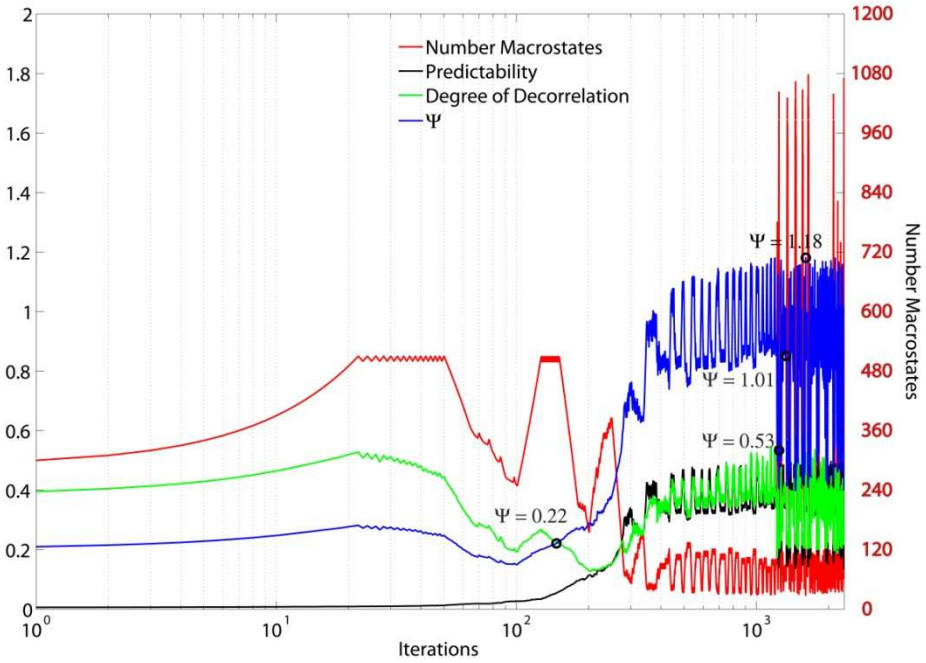

Figure 12: Objective function during an optimization run. The optimization run is characterized by a high degree of decorrelation values at the beginning of the optimization. This is due to the random distribution of Macrostates at the beginning of the optimization run. This high degree of decorrelation in the first hundred iteration steps governs the curve progression of Ψ. With further optimization steps the values of Ψ is dominated by the predictability values. Further the later optimization steps are characterized by an oscillatory behavior. The Ψ values corresponds to the state distributions shown Figure 12. The highest Ψ value is given by 1.18.

We investigated whether the objective function is influenced by the choice of motor parameters or maze type. Figures 14A and 14B show this influence on the average Ψ of optimized Macrostate configurations. There is a quite weak positive linear relation between Ψ and step-length (slope 0.0072 per microstate). Underlying this trend is a positive linear relationship between predictability and step-length (0.0059 per microstate) and a negative one between degree of decorrelation and step-length (-0.0115 per microstate). Moreover, Ψ decreases slightly when angular noise is increased (-0.012 per degree). Step-length noise interacts with step-length in that the only effect of step-length noise occurs at step-length 5. At this value, increased step-length noise leads to a marginal decrease of Ψ. All of these noise effects are largely determined by predictability, degree of decorrelation being not consistently affected by action noise.

In short, some motor parameters affect the Ψ value of optimized states. Here, the most notable effect is caused by an increase in angular noise. This is quite intuitive, as it strongly influences movement accuracy, ultimately limiting the degree of predictability which can be achieved. However, the effect size is small compared to the dynamic range of Ψ (see, e.g., figure 14B). Thus we conclude that the optimization process reliably generates state configurations with high Ψ values and is robust with respect to the choice of motor parameters.

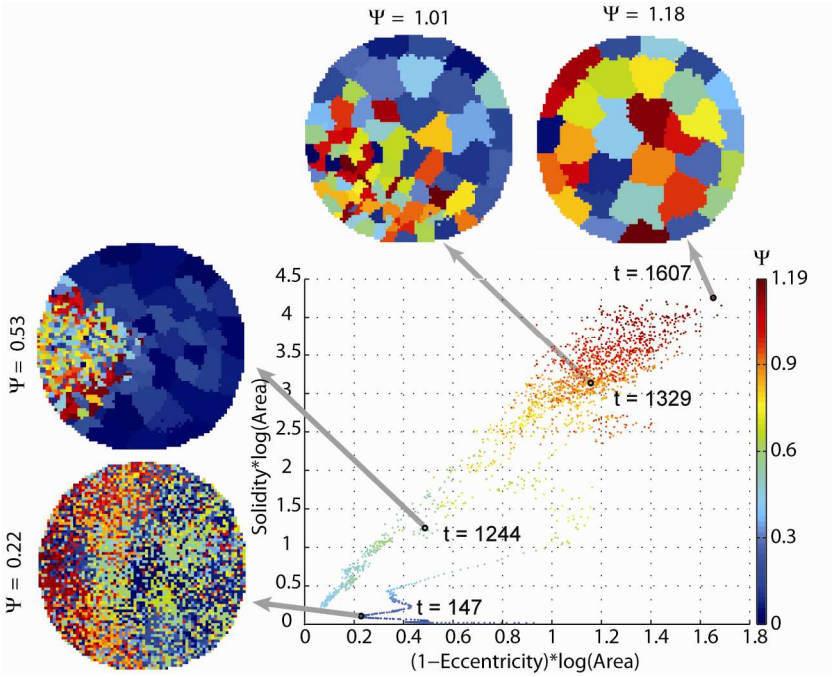

Figure 13: This figure illustrates the relation between three Macrostate properties: the objective function Ψ, size-weighted solidity and size-weighted roundness (1-eccentricity). Each point represents the average of these measures for one iteration step in an exemplary optimization run. The four insets show the spatial layout of the Macrostate configurations at selected iterations.

3.2.2 Spatial structure of Macrostates.

Each Macrostate occupies a two-dimensional region of the maze environment. Here, we analyzed how properties of these regions (see 3.1.4) change during the course of optimization. For the initial state configuration, the average Macrostate roundness (1-eccentricity) and solidity were 0.38 (± 0.02) and 0.012 (±0.006), respectively. The corresponding values for optimized Macrostate configurations were 0.66 (±0.08) and 0.89 (± 0.004).

This trend is illustrated in Figure 13 using an exemplary optimization run. Each dot corresponds to an iteration during this run. Color codes for the average Ψ value of the Macrostate configuration associated with that iteration while position represents size-weighted roundness and size-weighted solidity. (We weighted solidity and roundness by the logarithm of Macrostate area in order to prevent Macrostates consisting of single microstates from receiving high scores). It becomes clear from Figure 13 that high objective function values go along with large, round and compact states. Accordingly, Macrostate area, solidity and roundness increase during the optimization process. Note that an increase in Macrostate size is equivalent to a reduction of Macrostate number, as the size of the environment remains constant.

Next, we investigated how Macrostate spatial properties depend on the choice of motor parameters. Figures 14A and 14D shows that the area of Macrostates increases in response to an increase in step-length. In figure 14D we see that this effect interacts with angular noise σ: a larger σ leads to larger Macrostates. This interaction effect is to be expected from geometrical considerations. No such effect can be observed for step-length noise σ. Neither

roundness nor solidity of optimized Macrostates was affected by the choice of motor parameters. There were no conclusive effects of maze type on spatial properties.

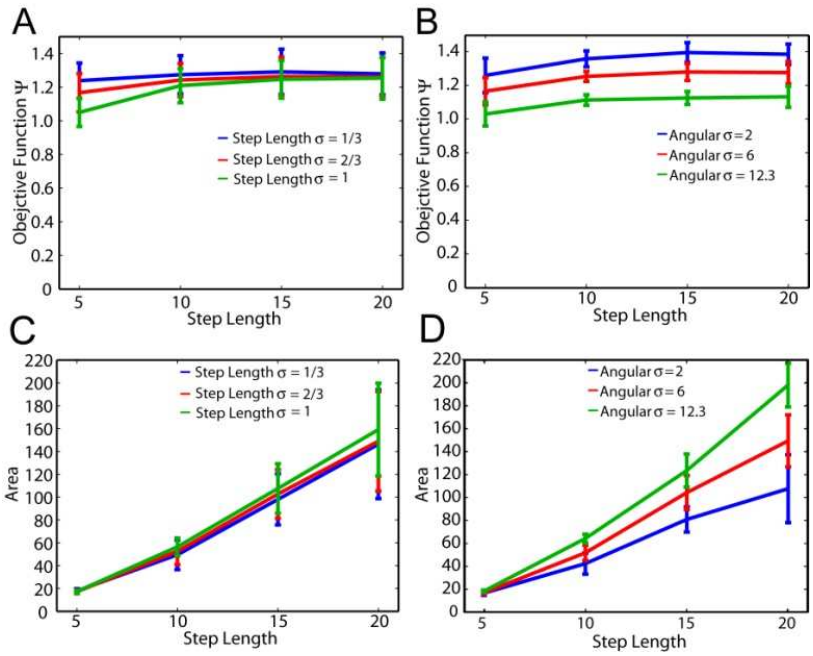

Figure 14: This figure shows the influence of motor parameter choice on the average Ψ and the average area of optimized Macrostate configurations. Here Step Length σ and Angular σ corresponds to the standard deviation of the noise added to the Steplength and to the angular orientation of the agent.

Next, we directly compared the topographical arrangement of optimized Macrostate configurations for different action parameters (see 3.1.1). We found that configurations with similar underlying step-lengths were most similar topographically (Figure 15). A comparable, yet weaker tendency was observed for angular noise. Step-length noise, on the other hand, was no consistent predictor of topographical similarity. Comparing Macrostate topography across maze types (same motor parameters, different maze) revealed that the Macrostate distributions of the square maze and the irregular square maze were more similar to one another (mean correlation across motor parameters 0.34 (± 0.06) than to the circular maze (circular vs. irregular 0.12 (± 0.02), circular vs. square 0.11 (± 0.02)).

We conclude that optimized Macrostate configurations possess large, spatially compact Macrostate distributions. Macrostate size is inversely related to motor precision: large-step-lengths in combination with large angular noise terms lead to large Macrostates. Finally, similar mazes lead to similar spatial distributions of Macrostates. We assume that it is this adaptation of Macrostate size and distribution to motor parameters and environment type which renders the objective function robust with respect to these issues. This makes sense intuitively, as coarsening the sensory space discretization maintains predictability in the face of increased motor noise.

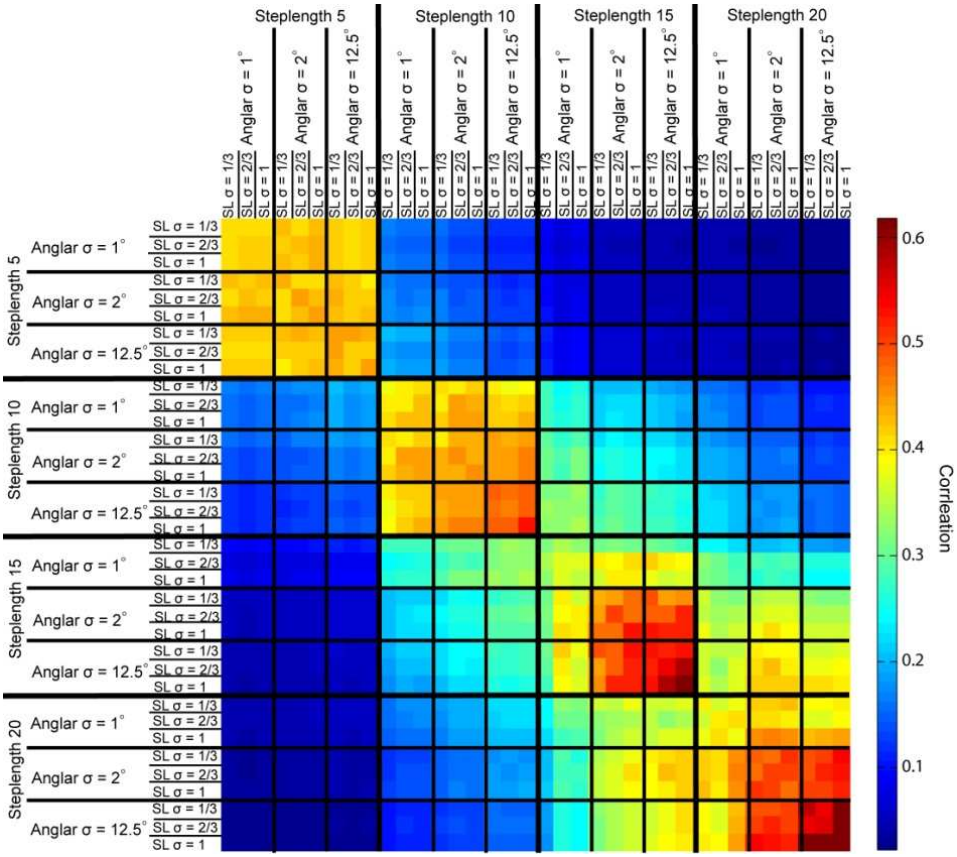

Figure 15: This correlation matrix shows the topographical similarity between optimized Macrostate configurations with different underlying action parameters. Here SL σ and Angular σ corresponds to the standard deviation of the noise added to the Steplength and to the angular orientation of the agent.

Here the maximum of Ψ appointed the Macrostate distribution of the optimization algorithm used for the analysis of the spatial structure. Ψ is linear combination of predictability and degree of decorrelation, weighted by the self connectivity. How do the weighting coefficients of predictability and degree of decorrelation influence these results? We analyzed this influence by varying the ratio of the weightings of predictability to the degree of decorrelation from 0.01 to 100. For each of these ratios the compactness and roundness of the optimal Macrostate distribution were analyzed, by averaging these values over all Macrostate within a Macrostate distribution. A cluster of very low compactness and roundness values were obtained, as shown in Figure 16A. These values correspond to Macrostate distribution with iteration steps smaller than 50 of an optimization process and were due to a high weighting of the degree of decorrelation and low predictability values as shown in Figure 16B. The high degree of decorrelation is caused by the random initialized Macrostate distributions, as shown in Figure 13. In contrast, increasing the ratio of the Ψ's components over a certain upper boundary does result in similar spatial properties. Further investigation showed that averaged over all optimization runs the mean upper boundary was 0.37 (±0.15). For all ratios larger than this upper boundaries did not resulted in a significant change of the predictability, degree of decorrelation, compactness and roundness of these Macrostate distributions. Thus, above a certain ratio of degree of decorrelation and predictability, no influence of the results was obtained.

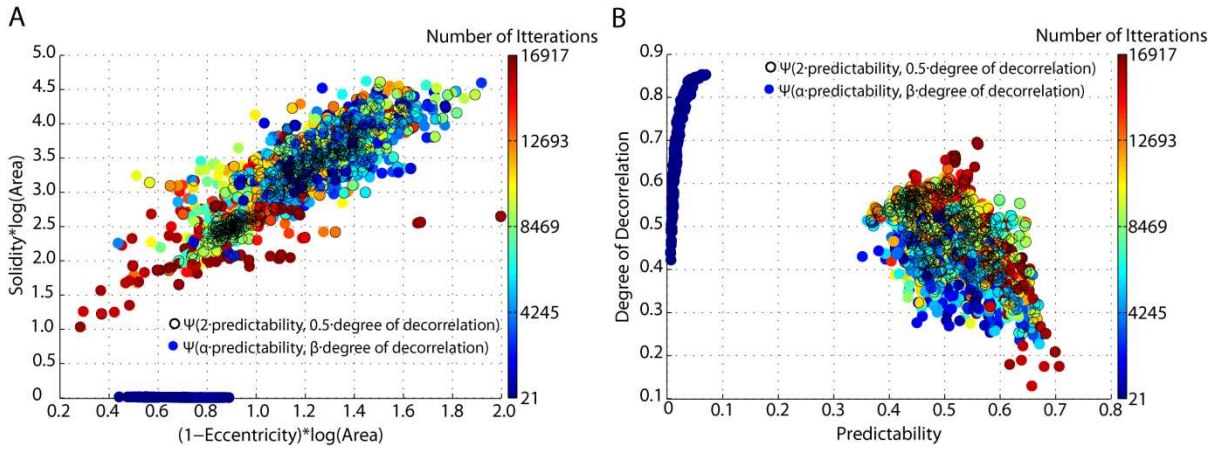

Figure 16: A. The eccentricity and solidity values of all possible optimal Macrostate appointed to the maximum Ψ, which here varies by different weighting of degree of decorrelation and predictability. The circle corresponds to Ψ, which was taken in the previous analysis (see Method section; the ratio was set to 4), while the other data points correspond to average spatial characteristics of an optimal Macrostate distribution with different weightings of Ψ. The color corresponds to the number of iterations of the optimization process these optimized Macrostate were obtained. B. Predictability values and degree of decorrelation for the different weightings of the components of Ψ.

3.2.3 Temporal coherence vs. predictability

Wyss and colleagues (Wyss et al., 2006) found that place-fields are a temporally stable neural representation of the video stream recorded by a robot moving in a two-dimensional environment. As both König and Krüger (König and Krüger, 2006) and Creutzig and colleagues (Creutzig and Sprekeler, 2008) pointed out that there is a connection between temporal stability and predictability, we compared the place-fields reported in (Wyss et al., 2006) to our Macrostates in terms of their temporal stability and state transition predictability.

To get from a continuous place-field activity distributions to a discrete partitioning of space, we applied a winner-take-all process to the activity distributions. The result was of the same form as the Macrostate distributions, which means that predictability could be computed on it once motor parameters had been selected. The resultant Macrostate distributions are shown in Figure 17B. To map discrete Macrostates to a continuous activity distribution, we first fit a two-dimensional Gaussian to the largest connected component of each Macrostate. After two standard deviations, the slope of this Gaussian was adapted to match the falloff behavior of the lower 5% percentiles of the activity distributions reported by Wyss and colleagues. This initial activity distribution associated with each Macrostate was then refined by adding small Gaussians to it until a winner-take-all operation yielded the original discrete Macrostate configuration. Once a Macrostate configuration had been mapped to a place-field configuration as shown in Figure 17A, its global stability could be assessed by averaging the stability objective with respect to each cell's activity (Wyss et al., 2006). Stability is a measure between zero and infinity, where zero would denote perfect stability. The reported values were normalized such that a random state configuration receives unit stability.

For the actual comparison we chose a place-field distribution reported in (Wyss et al., 2006) and a matching Macrostate configuration, shown in Figure 17A. The latter was obtained by applying our optimization algorithm to the sensory space of an agent whose motor parameters

were extrapolated from figure 13 given the area of the discretized place-field distribution from (Wyss et al., 2006). We computed ten optimized Macrostate configurations using these parameters. The optimization process yielded configurations consisting of an average of 23 Macrostates, which is close to the 25 place-fields reported by Wyss and colleagues.

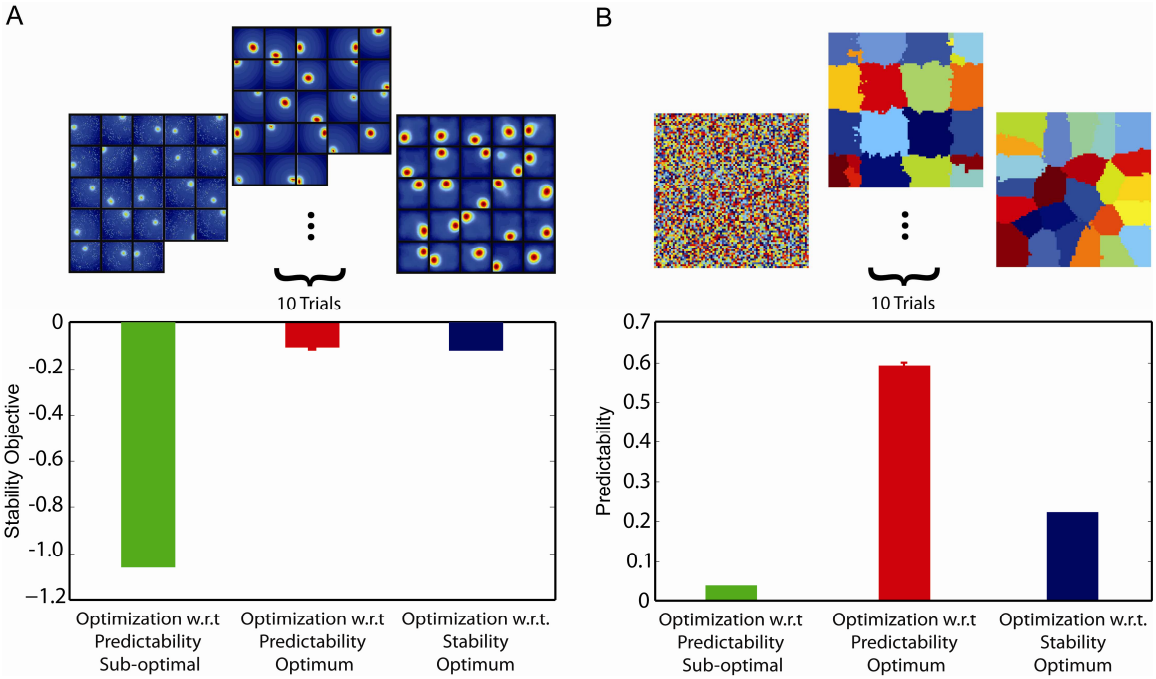

Figure 17: Comparison predictability and stability. A. Stability comparison: Here the stability objective (Wyss et al.) of the activity distribution was calculated. The left upper part shows approximated place cell activities obtained by a random distribution of Macrostates. The middle part exemplary place cells activities of a distribution of Macrostates, optimized for predictability. The stability objective for these optimized Macrostate distribution was averaged over 10 different optimization runs. The activity pattern on the right site was obtained by optimizing stability (Wyss et al., 2006). B. Predictability comparison. The predictability values were calculated of a random distribution of Macrostates (shown on the left side) and of 10 different optimal Macrostate distributions, optimizing predictability. On the right side a Macrostate distribution is shown, which resulted from optimizing stability.

The mean stability value of these ten Macrostate configurations was 0.0981 (± 0.00092), while the stability of Wyss et al.'s place-fields was 0.1111, shown in Figure 17A. To control for possible biases induced by the mapping from discrete Macrostates to continuous distributions, we discretized the reported place-fields via a winner-take-all operation and then turned them into continuous place-fields again using the same algorithm used to turn Macrostates into place-fields. Using this technique, Wyss et al's place-fields received a stability value of 0.1009. This means that our Macrostates are just as stable as states that were directly optimized for stability. The average predictability of the ten Macrostate configurations was 0.584 (± 0.003), while the predictability of Wyss et al's place-fields was 0.216, visualized in Figure 17B. In comparison, a randomly distribution of Macrostate resulted in a predictability value of 0.0437 and a stability value of 1.0761. Thus both optimization processes resulted in an increase of predictability and stability values compared to a random distribution, while optimizing with respect to predictability also leads to an increase in temporal stability, whereas the inverse is not the case.

3.3 Discussion

3.3.1 Summary of the results

The proposed optimization algorithm reliably generates sensory space representations with predictable and decorrelated state transition probabilities. The Macrostates constituting these representations are spatially compact and isotropic, and comparable to place fields. The algorithm is robust for a variety of agent motor parameters and the spatial characteristics of the optimal states are independent of the linear coefficients of the constituents of Ψ. The size and topographical distribution of Macrostates are adapted to the agent's motor capabilities, as well as to its environment. Further, by comparing Macrostates to the temporally stable place-fields reported by Wyss and colleagues (Wyss et al., 2006), we found that states optimized with respect to predictability are also temporally stable. These results suggest that the motor apparatus could play a profound role in the formation of hippocampal place-fields in the actively behaving animal. The optimization process is based on tools from graph theory and thus can be easily employed to optimize other sensory modalities.

3.3.2 Generality of the optimization process

Here we have made the claim that the algorithm generalizes over different modalities. This claim of generality is founded on the techniques used for the optimization process. This process utilizes tools used in graph theory, with the transitions representing edges and Microstates/Macrostates corresponding to vertices. Further, the definition of Microstates/Macrostates and transition probabilities does not involve any assumptions about the statistical characteristics of the sensory space or the agent's specific behavior (see 3.1.2). Additionally, the different transformational processes (cut and merge) use only tools related to the concepts of graph theory, again without using any assumptions about the underlying sensory space and agent's behavior. In summary, the presented algorithm does not enforce any particular interpretation about the modality of the sensory space or the agent's behavior, and thus has the potential to be applied to other sensory modalities and behavior.

The optimization algorithm is based on the behavioral interpretation of the sensory space. As noted in previous discussions (see section 2.3.2), the agent generates a topography by interacting with its environment, i.e. by learning the transition probabilities. Based on this topographical interpretation, the optimization reorganizes the sensory space in order to increase the predictability of the sensory outcome. Here we can hypothesize that such topographical interpretation would also account for any sensory ambiguities and would involve them in the low-dimensional interpretation captured by the Macrostates. Thus the optimization algorithm accounts for the complete complexity of the sensory state. However, further experiments must be done in order to interpret the Macrostates of these complex sensory spaces. In contrast, the presented thesis is based on a simplified sensory space, spanned by the agent's position within the environment. In order to expand this work to more complex sensory spaces like the visual space, we suggest a mapping between the positional sensory space and the visual sensory space used in Wyss et al. (Wyss et al., 2006) by taking an appropriate set of unique camera views and mapping them to the corresponding spatial position (see e.g. Franzius et al., 2007). Based on this mapping we hypothesized that a clustering of the visual events with a relevant spatial connotation could be achieved by the

presented optimization algorithm. As the algorithm only takes into account the transition probabilities between sensory states, the optimization algorithm can be applied to a variety of sensory modalities.

3.3.3 Comparison to reorganization algorithms

The optimal organization of the sensory space into states presented here shares some similarity with state aggregation in reinforcement learning. Reinforcement learning is concerned with how an agent ought to take action to solve a task (Sutton and Barlow, 1998). The goal of the task is related to a reward which is learned by means of the agent's interaction with its environment. In order to solve the task, this learning algorithm allows the agent to execute the action that relates to the maximization of the reward. One major component of reinforcement learning is the value function, which stores the reward associated with each state of the sensory space. Dynamic programming allows a mathematical formulation of these concepts and suggests an optimal solution for a given task. The solution to the Bellman equation provides an optimal solution to a given task, and corresponds to the value function for this task (Bellman, 1957). The different types of reinforcement learning represent different methods of approximating the value function.

In general this approximation is based on a division of the task-relevant sensory space into states. The choice of the states influences the characteristics of the value function, and thus the behavior of the agent to solve the task. State aggregation (Moore, 1991) considers the optimal choice of these states in order to solve the underlying problem. Moore suggested that a higher resolution of states should be chosen in regions of interest which are encountered while reaching the goal. In a more formalized approach, Singh and colleagues (Singh et al., 1995) suggested that states should be chosen in order to decrease the Bellman error, defined as the difference between the approximated value function and the exact solution of the Bellman equation. In contrast to our model, this study predefined the number of states. Similar to our work, Chrisman (Chrisman, 1992) suggested a choice of state that increases the predictability of the sensory outcome of the agent's actions. In this approach only a splitting of states is allowed, which reduces the number of possible state configurations compared to our algorithm. Further, McCallum (McCallum, 1993) reformulated this algorithm by using not the predictability of the sensory outcome, but the prediction of the reward, as a qualitative measure of the state configuration. In contrast to the previous approach, merging and splitting of states are allowed.

Similarly to predictive coding, Chrisman and McCallum's approaches also try to solve the problem of incomplete perception associated with perceptual aliasing (see 1.1.3.3). McCallum (McCallum, 1996) suggested an algorithm of state allocation for solving the problem of perceptual aliasing by using a decision tree, which creates states if they significantly contribute to an increase of the reward. Further, Reynolds (Reynolds, 2000) suggested that the boundaries of states should represent decision boundaries, which define regions of the sensory space where the same action maximizes the reward to reach the goal. In summary, most of the literature dealing with state allocation reorganizes the sensory space in order to increase an agent's performance in solving a particular task, and do not address the problem of sensory processing. State allocation chooses a state configuration in which a

reinforcement signal is used to optimize a particular behavior, while in our approach the optimal reorganization of the sensory space is due to the agent's possible scope of actions in this environment. Only Chrisman (Chrisman, 1992) suggested an organization of the sensory space in order to increase predictability, but restricted the possible number of states by allowing only one process (splitting) to change the state configuration. In contrast to state allocation, our algorithm acts on the sensory space with all its motor capability and is not bound to using a particular behavior to reach a goal.

3.3.4 Principles of the hippocampus

In the present study we associated the Macrostates with the place fields of place cells. Because place cells are a particular feature of the hippocampal region, here we give a short overview of their physiological basis and recent theories of hippocampal functionality. We will also discuss the sensitivity of place cells to sensory and self-motion features, and relate this to the reorganization of sensory space in our model. Although the hippocampus has been thoroughly investigated and its physiology is well known, consensus has not yet been reached regarding the underlying functionality of the hippocampus and its neuronal properties.

The hippocampus in human beings is located in the medial temporal lobe and receives highly processed information from all sensory modalities as well as multimodal regions via the entorhinal cortex (EC). Information flows in a roughly unidirectional fashion from the EC to the dentate gyrus (DG), then to hippocampus and then to the subiculum and back to the EC. These structures constitute the hippocampal region, which is similar in a variety of different mammals. In addition to this basic pathway of neuronal signals, many direct connections between these structures can be found (a detailed discussion of the structure can be found in Gluck and Myers, 1997). Further, the morphological structure of the hippocampus is divided into two main subregions, CA1 and CA3. In contrast to CA1, the neurons in the CA3 region are characterized by high interconnectedness and receive input via the mossy fibers from the DG, while CA1 receives input from CA3 and EC.

Early in the research, it was found that the hippocampus plays an important role in memory. Scoville and Millner (Scoville and Millner, 1957) reported that a surgical destruction of the hippocampus resulted in the patient's inability to remember events after the surgery. Further investigations revealed hippocampal involvement in encoding, consolidation and retrieval of episodic, semantic and spatial memories (Marr, 1971; O'Keefe and Nadel, 1978; Squire, 1992; McNaughton and Morris, 1987; for review Eichenbaum, 2000). The involvement of the hippocampus in memory processing is independent of sensory modality, and thus the hippocampus represents a processing step in a multimodal memory system (Gluck and Myers, 1997).

Different models evolved to capture the above-mentioned properties of the hippocampus. Marr (Marr, 1971) was one of the first to suggest that auto-associative models might capture some of the hippocampal properties. In this model, memories are represented by patterns of active neurons, with two different processes: storing and recall of memories. The storing procedure involves a pattern separation process relating to the ability to differentiate between two different memory events. Based on the physiological properties of the CA3 region of the

hippocampus, Marr hypothesized that the function of the hippocampal region can be characterized as a pattern separation process (a more detailed review: Gluck and Myers, 1997). Further, it is hypothesized that different structures of the hippocampal region are involved in this pattern separation process. For example O'Reilly (O'Reilly et al., 1994) assumed that DG increases the sparseness of the activity from EC and enhances the pattern separation ability of CA3. Further investigations characterized the functions of DG as spatial pattern separation while CA3 shares some similarity with temporal pattern separation. In summary, the hippocampus is assumed to be involved in pattern separation, which was first validated by Gilbert and colleagues (Gilbert et al., 1998).

The recall of memories is also often described as pattern completion. Pattern completion is a characteristic feature of episodic memory – for example the event memory of a meal might include associations about the food that was eaten. Thus, pattern completion involves the activation of a part of a pattern, which should then reactivate the whole corresponding stored pattern. This process is captured in the attractor dynamics of the autoassociative neural network (Marr, 1971; McNaughton and Morris, 1987; Treves and Rolls, 1992; 1994). Here the stored neural activity patterns can be interpreted as attractors, such that an external input prompts the neural activity to evolve dynamically and approach one of these attractors, usually the one most closely correlated with the input. These neural attractor-based networks are characterized by a sharp and coherent transition to one pre-established basin of attraction while the inputs vary progressively. In summary, autoassociative models of memory capture the pattern separation process in attractor dynamics of memory recall, which is assumed to take place in the hippocampus.

3.3.5 Neuronal properties of place cells

Theories about the involvement of hippocampus in spatial memory were strongly supported by the discovery of place cells in the CA1 and CA3 regions. Place cells were first reported by O'Keefe and colleagues (O'Keefe and Dostrovsky, 1971) and their neural response properties are sensitive to the location of the animal in an environment. The place cell's place field characterizes the region of the environment in which a cell is active. It is assumed that these place cells constitute a cognitive map of the environment. In the next section we will discuss different characteristics of the formation of place cells and the causal relation of sensory as well as self-motion features which are involved in their formation.

Wills and colleagues (Wills et al., 2005) tried to validate the attractor characteristics of autoassociative network models by investigating the remapping characteristics of CA1 place fields. Remapping refers to the ability of place cells to encode different locations in different environments (Bostock et al., 1991), while being stable in the same environment during different trials. The attractor characteristics were validated by correlation of the place fields during continuous changes of the environment (Wills et al., 2005). Further similar experiments were done using recordings in CA3 (Leutgeb et al., 2005), which revealed a hysteresis effect of CA3 place fields. In summary, the attractor characteristics of autoassociative models have indeed been found and validated in neurons of the hippocampus.

As mentioned above, there are physiological differences between the CA1 and CA3 regions of hippocampus, but does this difference in connectivity also mean there are differences in the place cells of these regions? In general no major quantitative differences, for instance in the size of place fields, were found (Barnes et al., 1990). Leutgeb and colleagues (Leutgeb et al., 2004) investigated the remapping property of place cells in the context of changes in features of the environment, like the shape of the arena or the surrounding room. Remapping in CA3 was obtained by any change of either the arena in which the agent is navigating, or the room in which the arena is situated, and thus involves different visual cues, while place cells in CA1 were relatively stable with a change obtained only by varying both parameters. These results support the conclusion that CA3 identifies context, while CA1 place cells can respond to individual landmarks. Based on the study of Shapiro (Shapiro et al., 1997), which highlighted that place cells respond to particular cues in an environment, Lee and colleagues (Lee et al., 2004) tried to differentiate the properties of the place cells of both hippocampal regions on the basis of distal and proximal cues. They hypothesized that CA3 place cells are dominated by local cues while CA1 place cells are modulated by a mixture of distal and local cues. However, as stated by Lee et al. more investigations are necessary in order to support this hypothesis (Lee et al., 2004). In summary, although no major differences between the place fields were found, it is assumed that place cells in CA3 identify the context of the environment, while CA1 place cells respond to individual landmarks.

The relevant features responsible for the formation of place cells have also been thoroughly investigated. In general, place fields are equally distributed over the whole environment and the topological order of place fields is not reflected in the topological order of the place cells in the hippocampus (O'Keefe et al., 1996; Redish et al., 2001). Yet, some place cells are sensitive to metric properties of the environment of the behaving animal, for instance the distance to the boundaries of the arena (Muller and Kubie, 1987; O'Keefe and Burgess, 1996; for review Jeffery, 2007). Further, many place cells are bound to one or a variety of visual, auditory, tactile and olfactory features of the environment (Shapiro et al., 1997). Not only sensory features play an important role in the determination of place fields, but also information from the vestibular system (Jeffery, 2007) and the motor system (Save et al., 1998; Terrazas et al., 2005) plays a crucial role. Many studies have suggested the importance of multisensory environmental features, as well as self-motion features, in the formation of place fields.

In recent years, investigations of EC have provided new insights into the formation of place fields. Fyhn and colleagues (Fyhn et al., 2004) found cells in EC with similar properties to place cells in the hippocampus. These cells have the same information content regarding the location of the animal as place cells in the hippocampus, but the cells in EC had different place fields. Further research on EC resulted in the discovery of boundary cells, cells firing at the boundary of an environment (Savelli et al., 2008). However, the most outstanding finding in the EC – more precisely in the medial entorhinal cortex (MEC) – is that of grid cells (Hafting et al., 2005). Grid cells are cells with a periodic firing pattern that has a spatial receptive field similar to a 2 dimensional grid, such that these cells are active whenever the position of the animal coincides with a vertex of a grid of equilateral triangles spanning the

surface of the environment. These activation patterns are also independent of the shape or size of the environment (Hafting et al, 2005; Fyhn et al., 2005). Thus, these results suggested that gird cells in the MEC provide metric information about the animal's environment. Although the underlying computational procedures involved in the generation of grid fields are not known, it is assumed that path integration plays a crucial role (For further discussions McNaughton et al., 2006). It is assumed that these gird cells, which represent the metric information of the environment, have an impact on the formation of place cells (McNaughton et al., 2006).

The hippocampus contains further neurons with different response properties like head direction cells, also known as view-directed cells (Taube et al., 1990). These neurons fire depending on the orientation of the animal within the environment. Each of these head direction cells has a preferred orientation, to which it responds with maximum firing rate. Different theories have been put forward regarding the involvement of head direction cells in the formation of place cells. Up to now, neither the underlying mechanisms of the formation of the response properties of these head direction cells, nor the interactions between these cells and place cells are fully understood.

In summary, different experiments have provided insights into the formation of place cell response properties by means of different environmental features (like landmarks), metric information and input from the motor system. Further, it has been suggested that grid cells (reviewed by McNaughton et al. 2006) or head direction cells have an impact on the formation of place cell response properties, but this influence is still an open question. Thus, although place cells have been thoroughly investigated, the results from the different physiological experiments have not resulted in a consistent theory defining which kinds of information from different sensory systems and the motor system are crucial for the formation of place cell response properties.

3.3.6 Models of hippocampal place cells

Based on physiological findings, two modeling approaches for the formation of hippocampal place cell response properties have been suggested. The first one involves the anatomy and the underlying connectivity of the hippocampus and hippocampal region (Stringer et al., 2004; Strösslin et al., 2005; Brunel and Trullier, 1998); the second is based on statistical properties inherent in the visual input (Wyss et al., 2006; Franzius et al., 2007). Regarding the latter sensory-driven approaches, both Wyss and Franzius applied the general coding principle of temporal coherence in a hierarchical network with visual input from a behaving robot. This was done without any consideration of the neural circuitries of the hippocampus or the connected sensory cortices representing the ventral visual pathway. Nevertheless, these studies derived the complex sensory representation of place cells. The success of these studies suggests that the neural circuitries on the different levels of the ventral visual pathway are computational similar. They further suggest that place cells do not need any explicit object recognition to extract spatial information; this information can be extracted directly from the statistics of the perceived natural stimuli. To my knowledge, Franzius and Wyss are the only studies which derived different neural properties like place cells or head direction cells from the statistical properties of the sensory input. Because these approaches utilize a

general coding principle similar to our approach, we will give a short discussion of the work of Wyss and Franzius followed by a comparison of their work to ours.

The importance of multisensory information for the formation of place cells has already been mentioned above. Save and colleagues (Save et al., 1998) stated that not only the visual system provided sufficient spatial information to form place cells, but also other sensory modalities (i.e. tactile, olfactory). As previous studies highlighted the validity of temporal coherence as a coding principle over different modalities, we hypothesize that the coding techniques applied to the visual sensory input in Wyss and Franzius generalize over the modalities and may derive neural properties equivalent to place cells based on information contained in non-visual sensory input. Thus, it could be claimed that the studies of Wyss and Franzius generalize over modalities.

Both Franzius and Wyss used the passively perceived visual input of a behaving agent in an environment. Yet, the temporal statistics of the visual input are determined by the agent's behavior. Franzius and colleagues highlighted that the temporal characteristics of the visual input had an impact on the formation of place cells, thus highlighting the impact of the agent's behavior on the sensory-driven formation of place cells. As a corollary of the generality of the normative coding of principle temporal coherence, it seems likely that place cells can also be derived from the input of other, non-visual modalities. In summary, the work of Wyss and colleagues (Wyss et al., 2006; and the similar work of Franzius et al., 2007) could model neural response properties of place cells from the statistical characteristics of the sensory input generated by a behaving robot.

3.3.7 Involving actions in models of place cells

In contrast to models of sensory-driven place-cell formation, we highlighted that place cells can additionally be interpreted as an optimization of the sensory space by increasing the predictability of the sensory outcome of an agent's action. Expanding on the work of Wyss and colleagues, the optimization scheme presented here combines sensory as well as motor components.

Terrazas and colleagues (Terrazas et al., 2005) investigated the influence of an animal's behavior on the formation of place fields by self-motion features. This study compared place fields of actively behaving rats to place fields formed by passive stimulation with the same visual input (i.e. of rats being passively driven along identical paths of the freely behaving rat). Even though the velocity characteristics of the movements and hence also the temporal characteristics of the visual input were similar in both conditions, the place fields of actively-perceived sensory input contained more information about the animal's position within the environment than place fields derived under passive sensory input. This demonstrates the involvement of motor or self-motion signals in the formation of place cells in addition to the motion information captured by the temporal characteristics of sensory input.

Here, we suggest that the motor system is involved in the formation of place cells, and extend the work of Wyss and colleagues by arguing that motor signals act directly on the sensory space and allow the optimization of the predictability of the sensory space. This assumption

allowed the number of states to be given by the agent's motor capability and the environment, while in the work of Wyss and colleagues the architecture of the neural network defined the number of states.

Previous studies proposed an involvement of the motor system via path integration processes (Etienne and Jeffery, 2004; for a review see McNaughton et al., 2006). Path integration refers to the continuous integration of self-motion features into a representation of directional distance, yielding a continually updated representation of direction and distance from a certain starting point.[4] Integration, in combination with external cues like visual landmarks, has been thought crucial for the formation of place cells (Etienne and Jeffery, 2004, for a review see McNaughton et al., 2006). In contrast, we state that actions, via their representations in motor areas, act directly on the sensory space in order to reorganize this sensory space by increasing the predictability of the action's sensory outcome. Hence, here we do not use any computational processes (like path integration) for the extraction of self-motion features in order to reorganize the sensory space in place fields. Thus, in contrast to an *interaction* of motor signals with sensory spatial features via computational processing of this motor signals, we suggest a direct involvement of the motor components in the *formation* of a complex, low-dimensional, sensory representation of space such as place cells.

3.3.8 Discussion of concepts and techniques of reorganization

In this section we discuss techniques as well as concepts used in the reorganization algorithm. We shortly outline the first steps of a biologically motivated implementation of the rule-based heuristics of the optimization process. A brief discussion of unsupervised learning is given, and we will shortly discuss online learning vs. batch learning, which was used here. Following this, we evaluate whether our algorithm can capture neural response invariance properties, which are similar to the general coding principles discussed in the Introduction. Next we shortly relate the reorganization of the sensory space to sensorimotor space and discuss the comparison between the coding principles of predictability and temporal coherence.

For reasons of biological plausibility, we must consider how the split and merge processes of the rule-based heuristics of the optimization process could be biologically implemented. For example, Einhäuser and colleagues formulated the coding principle of temporal coherence in a neural network and further suggested a biologically inspired implementation (Einhäuser et al., 2002). Because artificial neural networks share some similarities with real neurons, a reformulation of the reorganization process in such a framework would be an appropriate first step towards a biologically inspired implementation. Further, the optimization of predictability has to be translated into a biologically inspired learning process like autoassociative learning (Hebbian learning), which is often involved in neuronal learning processes. Thus a reformulation of this reorganization process within the framework of neural networks and autoassociative learning would be a step towards a biologically plausible implementation.

[4] Navigation based on these features is referred to as 'dead reckoning' (Wehner and Sirivansan, 1981)

The presented optimization process reorganizes the sensory space in an unsupervised manner to increase the predictability of the sensory outcome of performed actions. In general, unsupervised learning subsumes learning procedures that exploit the statistical regularities of data. Because sensory coding aims to relate neural response properties to their sensory input, unsupervised learning is a useful tool to investigate this relation and has indeed been successfully applied to natural stimuli to understand different processing steps of the visual system (Einhäuser et al., 2002). Here we compare the unsupervised learning procedure applied in our reorganization process with that of the studies mentioned. These studies implement unsupervised learning in neural networks and employ a gradient descent or ascent method that maximizes certain properties of neuronal activity. That is, the gradient descent method is used in combination with an objective function. This objective function represents a mathematical formulation of certain statistical properties of the neural activity distribution which should be maximized during learning. The gradient ascent method applied to the objective functions ensures that the weights of a neural network will be iteratively adjusted such that the objective function reaches a maximum, be it local or global. The learning procedure is terminated when the maximum of this objective function is reached. In contrast, our algorithm is guided by rule-based heuristics informed by predictability and the degree of decorrelation. Thus, this algorithm does not truly perform gradient ascent on the objective function. Instead, it may pass this maximum and proceed to produce sub-optimal reorganizations of a sensory space from then on. In practice, this is indeed the case – increasing the number of iterations did not yield higher values of the objective function. Furthermore, up to now, we did not compute the upper boundary of possible objective function values for a certain set of motor parameters and a certain environment. Approximating this boundary would provide a useful termination criterion. In summary, our algorithm successfully increased the predictability by changing the sensory organization in an unsupervised manner, but to improve the optimization process, a termination criterion has to be provided.

Most studies applying general coding principles in neural networks are capable of learning in an online fashion, while our optimization process is based on batch learning. Online learning involves the use of current sensory input to adjust the weights in neural networks. Thus, the values of the connecting weights result from the processing of previously and currently perceived sensory input. In contrast, our optimization process utilized transition probabilities resulting from a simulated agent learning these probabilities, by executing an exhaustive exploratory motor program. Thus, in an exploration phase the transition probabilities between the sensory events are learned, followed by a reorganization process based on the previously learned transitions. This fashion of learning is defined as batch learning, and separates the exploration and reorganization of the sensory space. Here this division was chosen to reduce the number of parameters of the optimization process, such as the number of experiences needed to generate an optimal organization; and to restrict the time needed to obtain an optimized organization of the sensory space. Finally, to modify the underlying optimization scheme to the biologically plausible neuronal space, the exploration and reorganization stages have to be merged, such that the reorganization is based on the transition probabilities calculated by the current number of explorations in an online-learning scenario.

A major hypothesis of sensory coding is to relate invariant neural response properties to their sensory input. Is the presented algorithm equally able to derive invariant neural response properties from a reorganization of the sensory space? Temporal coherence, a general coding principle, explains invariant neural response properties such as the invariance of place cells to the agent's orientation at the same position, or the phase invariance of complex cells from natural visual stimuli. In this study we compared clusters in the positional sensory space with place cells. Thus, in contrast to the work of Wyss or Franzius (Wyss et al., 2006; Franzius et al., 2007), we here predefined the orientation invariance of place cells by the choice of the positional sensory space in order to test our algorithm in a simplified paradigm. This gives rise to the question of the ability of the presented reorganization process to capture invariance from sensory input, similar to the work of Wyss and Franzius (Wyss et al., 2006; Franzius et al., 2007). Thus, we will discuss whether our optimization results in orientation-invariant clusters from visual sensory input, as done by Wyss and colleagues (Wyss et al., 2006). Here we applied the optimization scheme to a navigational paradigm with a sensory space defined by our agent's position, which does not entail any sensory information of the agent's orientation. In order to investigate the ability of our reorganization algorithm to yield this orientation invariance we have to change to the visual sensory space, which allows a differentiation between orientations. Orientation invariance according to our framework relates to clusters of sensory visual events coming from different orientation of the agent at the same position. Additionally, as seen previously, our reorganization algorithm clusters sensory events with similar transition probabilities. In order to achieve orientation-invariant clusters, the motor capabilities have to be expanded by rotation of the robot at the same position. We would expect the algorithm to cluster sensory events according to different orientations, with similar transition probabilities caused by rotation and thus deriving orientation invariance. In summary, here we suggest that depending on the agent's motor capabilities the algorithm is able to capture sensory invariance.

In this thesis we claimed to reorganize not only the sensory space but the sensorimotor space. We reorganized the sensory space by involving the agent's action capabilities. This reorganization process is based on the effect of the agent's motor capability on the sensory space, captured by the sensory state transitions learned by the agent's interaction with its environment. Yet, in our optimization process we differentiated between the motor component and the sensory component, in contrast to the definition of the sensorimotor space spanned by these components. Nevertheless, since our optimization procedure accounts for the effect of the motor capabilities on the agent's sensory space, this optimization process is driven by the statistical properties of the sensorimotor space. Hence, our reorganization scheme accounts for the sensorimotor space.

The relation between the general coding principles of predictability and temporal coherence has already been investigated in different approaches. In the context of predictive coding (see 1.1.3.3), Creutzig and colleagues (Creutzig et al., 2008) proved that the solutions of Slow Feature Analysis are contained in the predictive coding principle. Because Slow Feature Analysis and the temporal coherence approach are similar approaches (see 1.1.3.2), it follows that predictive coding entails temporal coherence. Further, König and Krüger (König and

Krüger, 2006) suggested that temporal coherence can be interpreted as 0th order predictability, because temporally coherent sensory input does not change on a given time scale and thus is also highly predictable. Further, they state that the predictability principle would extract more features than temporal coherence, as it also predicts future sensory input. Additionally we investigated the relation between predictability and temporal coherence by comparing our resultant low-dimensional representation with the place fields obtained by Wyss and colleagues (Wyss et al., 2006). It has to been taken into account that the two studies were applied on different sensory spaces. While Wyss's temporal coherence principle was applied to the visual input, our predictability approach reorganized the positional space. Moreover, in Wyss's temporal coherence approach all visual information was taken into account during the movement of the robot, while in our predictive approach only the sensory outcome of the action was taken into account. Further, the temporal coherence approach used a hierarchical network to extract slowly-varying visual features on different time scales; while in our model, based on action prediction, one timescale suffices to define the execution of an action. In order to compensate for these differences, we compared the results of both approaches in the positional domain of the environment. This comparison was done by mapping the place cells' activity to clusters, in a winner-take–all process. Optimizing temporal coherence leads to local place cell activity that is high only in a local region of the environment, and hence, temporal coherence maximization yields compact clusters. In contrast, maximizing predictability according to the motor capabilities defines the cluster distribution and thus their form and location. Further, in the predicative case the low-dimensional representation was highly predictive and temporally coherent, while for temporal coherence optimization, only temporal coherence was maximized. Despite the many differences between the temporal coherence and our predictability approach, consistently to other studies we proved that predictability entails temporal coherence and thus predictability constitutes a more general coding principle.

In summary, for reasons of simplicity we introduced a heuristic rule-based optimization process, which successfully increase the predictability and degree of decorrelation of the current state configuration. In future work, we would like to change the optimization process to be more biological plausible. For biological plausibility reasons we would adapt the algorithm to techniques mostly used in computational neuroscience. Thus we would modify the optimization process by implementing this process in a neural network combined with autoassociative learning in an online learning scenario and a termination criterion for the optimization procedure. Further we suggest that the presented optimization process is able to capture invariances of sensory representations by utilizing the general coding principle of predictability.

4. Outlook

In this thesis we developed an approach to reorganize an agent's actively perceived sensory input and thus its sensorimotor space. In order to achieve such a reorganization of the sensory space, we first introduced a model of behavior, which has the potential to be generalized to different tasks and agents. Second, we introduced a heuristically rule-based optimization which reorganizes sensory spaces by increasing the agent's predictability of the sensory outcome of its actions. This reorganization scheme can be expanded to different sensory modalities, resulting in a low-dimensional sensory representation characterized by entities of sensory events. Further, this low-dimensional representation can be used to model behavior in an environment. Here we applied this reorganization process to a navigational paradigm, resulting in clusters of sensory events which have similarities to the response properties of hippocampal place cells. Hence, the presented reorganization process is able to capture neural response properties like place fields and the corresponding place cells can be interpreted as the result of an interaction of the sensory input and the motor capabilities of the agent.

What are the aims of future work in this direction? As well as the suggested modifications to the presented optimization process (see section 3.3.8), future research could apply the optimization process to more complex sensory spaces. In this study the optimization scheme was applied to a simple sensory space, with each sensory event corresponding to the agent's position within the environment. In such a sensory space each sensory event represents a unique event; in contrast, most sensory spaces in biological systems display a lot of ambiguities in their statistical properties. Hence, future research should investigate the effect of these ambiguities on the introduced optimization process. Moreover, the influence of a higher dimensionality of the sensory space on the reorganization process should be further investigated, which could result in more complex sensory representations. In summary, future work would be concerned with an investigation of the results of the optimization process applied to more complex sensory spaces.

This underlying idea of this thesis is that complex sensory representations can be understood as result of a reorganization of sensorimotor space. Although recent studies highlight the importance of combining action and sensation to understand aspects of the computational processes of the brain, we have yet to show that the presented reorganization scheme is implemented, or even implementable, in the neural circuits of the brain.

A further topic of this thesis was to highlight the potential of predictability as a general coding principle. To emphasize the ability of general coding principles to capture different complex sensory representation, we would additionally investigate the statistical characteristics of an agent's sensorimotor space in relation to the general coding principle of predictability. This sensorimotor space would be spanned for example by visual sensory input and motor activation. Here the motor activation would correspond to a pattern of neural activity unique to each executable action of the agent. Such a sensorimotor space contains all the complexities of a behaving agent and such an investigation could result in new insights into the development of complex neuronal response properties. In summary, in a different line of research investigating the impact of the sensorimotor space on the development of neural response properties, we suggest it may be fruitful to examine the statistical

characteristics of the sensorimotor space according to predictability, and to relate these characteristics to the neural response properties.

The presented optimization process goes a step beyond current sensory coding approaches. While sensory coding incorporates only sensory information, here we incorporate information from the sensory and motor systems. Further, we have demonstrated that neural response properties can be explained by reorganizing the sensory space to increase the predictability of the sensory outcome of an agent's actions. In other words, only behaviorally relevant sensory information is important, with relevance defined by the capability to predict sensory outcomes of actions. Based on these concepts, the reorganization process can be seen to result in entities of sensory events, which could be hypothesized to be similar to symbols (Honovar and Uhr, 1994). Additionally, the neurons of the brain develop their response characteristics through the interactions between situated agent and environment. Based on this characteristics, here we hypothesize that taking the sensory as well as the motor system into account in combination with the normative coding principle of predictability could give rise to a profound understanding of complex neuronal representations, such as the specific responses of neurons in visual system (Kreiman et al., 2000). Thus, this hypothesis is similar to previous suggestions (Steels, 2003) that action and perception are important for the development of different categorical units (symbols), which are supposed to be the bases of higher cognitive processing like planning or perceptual learning (Ashby and Maddox, 2005). Further, especially in the scientific subfield of Artificial Intelligence, symbols and their corresponding syntax have been intensively investigated as a means of capturing complex cognitive processing like reasoning (Bader et al., 2008; Hitzler et al., 2004). These investigations could be newly approached on the basis of such a sensory reorganization process. In summary, combining sensory and motor information in combination with predictability could give rise to a profound understanding of neural response properties, as well as higher cognitive processes like planning.

5. References

ALOIMONOS, John ; SHULMAN, David A.: *Integration of visual modules: an extension of the Marr paradigm.* Boston [u.a.]: Academic Press, 1989. - 0-12-053020-1

ALEXANDER, H.W.; SPORNS, O.: An Embodied Model of Learning, Plasticity and Reward. *In: Adaptive Behaviour* (2002) 10: 143-159

ARLEO, A. ; GERSTNER, W.: Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. *In: Biol Cybern* (2000) 83(3):287-99

ASHBY, F.G. ; MADDOX, W.T.: Human category learning. *In: Annu Rev Psychol* (2005) 56: 149-78

ATICK, J. J.: Could information theory provide an ecological theory of sensory processing? *In: Netw. Comput. Neural Syst.* (1992) 3: 213-51

ATICK, J. J. ; REDLICH, N. A.: What does the retina know about natural scenes? *In: Neural Comput.,* (1992), 4(2):196-210

BACH, J.: The MicroPsi Agent Architecture. *In: Proceedings of ICCM-5, International Conference on Cognitive Modeling, Bamberg, Germany* (2003):15-20

BACH, J.; VUINE, R.: Designing Agents with MicroPsi Node Nets. *In:* Proceedings of KI 2003, Annual German Conference on AI. LNAI 2821, Springer, Berlin, Heidelberg. (2003): 164-178

BACH-Y-RITA, P.: *Brain mechanisms in sensory substitution.* New York: Academic P., 1972. - 0-12-071040-4

BACH-Y-RITA, P.: Tactile sensory substitution studies. *In: Ann N Y Acad Sci* (2004) 1013: 83-91

BADDELEY, R. ; ABBOTT, L. F. ; BOOTH, M. C. ; SENGPIEL, F. ; FREEMAN, T. ; WAKEMAN, E. A. ; ROLLS, E. T.: Responses of neurons in primary and inferior temporal visual cortices to natural scenes. *In: Proc Biol Sci* (1997) 264(1389):1775-83

BADER, S.; HITZLER, P.; HÖLLDOBLER, S.: Connectionist Model Generation: A First-Order Approach. *In: Neurocomputing* (2008) 71:2420-2432.

BARLOW, H.: Possible principles underlying the transformation of sensory messages. *In: Sensory Communication, W. A. Rosenblith, Ed. Cambridge, MA: MIT Press,* (1961) : 217—234

BARLOW, H. B.; FOLDIAK, P.: Adaptation and decorrelation in the cortex. *In: The Computing Neuron, ed. R Durbin, C Miall, G Mitchinson* New York: Addison-Wellesley (1989) 4:54-72

BARLOW, H.: Redundancy reduction revisited. *In: Network* (2001) 12(3): 241-53

BARNES, C. A. ; MCNAUGHTON, B. L. ; MIZUMORI, S. J. ; LEONARD, B. W. ; LIN, L. H.: Comparison of spatial and temporal characteristics of neuronal activity in sequential stages of hippocampal processing. *In: Prog Brain Res* (1990) 83:287-300

BECKER, S.: Implicit learning in 3D object recognition: the importance of temporal context. *In: Neural Comput* (1999) 11(2): 347-74

BELL, A. J. ; SEJNOWSKI, T. J.: The 'independent components' of natural scenes are edge filters. *In: Vision Res* (1997) 37(23): 3327-38

BELLMAN, R.: *Dynamic programming.* Princeton: 3. Aufl. Princeton Univ. Press, 1957.

BERNSTEIN, N.: *The co-ordination and regulation of movements.* Oxford [u.a.]: 1. Aufl. Pergamon Press, 1967

BIALEK, W. ; NEMENMAN, I. ; TISHBY, N.: Predictability, complexity, and learning. *In: Neural Comput* (2001) 13(11):2409-63

BLANCHARD, M.; VERSCHURE, P. F. M. J.; RIND, F. C.: Using a mobile robot to study locust collision avoidance responses. *In: International Journal of Neural Systems* (1999), 9:405-410

BLUM, K. I. ; ABBOTT, L. F.: A model of spatial map formation in the hippocampus of the rat. *In: Neural Comput* (1996) 8(1): 85-93

BOLZ, J. ; GILBERT, C. D.: Generation of end-inhibition in the visual cortex via interlaminar connections. *In: Nature* (1986) 320(6060): 362-5

BOSTOCK, E. ; MULLER, R. U. ; KUBIE, J. L.: Experience-dependent modifications of hippocampal place cell firing. *In: Hippocampus* (1991) 1(2): 193-205

BROOKS, R. A.: New Approaches to Robotics. *In: Science* (1991) 253(5025): 1227-1232

BRUNEL, N. ; TRULLIER, O.: Plasticity of directional place fields in a model of rodent CA3. *In: Hippocampus* (1998) 8(6): 651-65

BURGESS, N. ; DONNETT, J. G. ; JEFFERY, K. J. ; O'KEEFE, J.: Robotic and neuronal simulation of the hippocampus and rat navigation. *In: Philos Trans R Soc Lond B Biol Sci* (1997) 352(1360): 1535-43

CAYWOOD, M.S. ; WILLMORE, B. ; TOLHURST, D.J.: Independent components of color natural scenes resemble V1 neurons in their spatial and color tuning. *In: J Neurophysiol* (2004) 91(6): 2859-73

CHALMERS, D.J.: Subsymbolic Computation and the Chinese Room. *In: The Symbolic and Connectionist Paradigms: closing the gap. ed: John Dinsmore, Lawrence Erlbaum Associates,* (1992) 25-48

CHAPMAN, B. ; STRYKER, M. P.: Development of orientation selectivity in ferret visual cortex and effects of deprivation. *In: J Neurosci* (1993) 13(12): 5251-62

CHRISMAN, L.: Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. *In:Proceedings of the Tenth National Conference on Artificial Intelligence* (1992)

CREUTZIG, F. ; SPREKELER, H.: Predictive coding and the slowness principle: an information-theoretic approach. *In: Neural Comput* (2008) 20(4): 1026-41

DAN, Y. ; ATICK, J. J. ; REID, R. C.: Efficient coding of natural scenes in the lateral geniculate nucleus: experimental test of a computational theory. *In: J Neurosci* (1996) 16(10): 3351-62

DAUGMAN, J. G.: Entropy reduction and decorrelation in visual coding by oriented neural receptive fields. *In: IEEE Trans Biomed Eng* (1989) 36(1): 107-14

DÖRNER, D.: *Bauplan für eine Seele.* Reinbek bei Hamburg: 1. Aufl. Rowohlt Verl., 1999. - 3-498-01288-6

DONG, D.W.; ATTICK, J.J.: Statistics of natural time-varying images. *In: Netw. Comput. Neural Syst.* (1995) 6:345-58

DUFF, A.; WYSS, R.; VERSCHURE, P. F. M. J.: Learning Temporally Stable Representations from Natural Sounds: Temporal Stability as a General Objective Underlying Sensory Processing. *In: Artificial Neural Networks – ICANN 2007* (2007) 4669:129-138

EDELMAN, G.M.: Learning in and from brain-based devices. *In: Science* (2007) 318(5853): 103-5

EICHENBAUM, H. ; DUDCHENKO, P. ; WOOD, E. ; SHAPIRO, M. ; TANILA, H.: The hippocampus, memory, and place cells: is it spatial memory or a memory space? *In: Neuron* 23(2): 209-26

EICHENBAUM, H.: A cortical-hippocampal system for declarative memory. *In: Nat Rev Neurosci* (2000) 1(1): 41-50

EINHÄUSER, W. ; KAYSER, C. ; KÖNIG, P. ; KÖRDING, K.P.: Learning the invariance properties of complex cells from their responses to natural stimuli. *In: Eur J Neurosci* (2002) 15(3): 475-86

EINHÄUSER, W.; HIPP, J.; EGGERT, J.; KÖRNER, E. ; KÖNIG, P.: Learning viewpoint invariant object representations using a temporal coherence principle. *In: Biol Cybern* (2005) 93(1): 79-90

ETIENNE, A.S. ; JEFFERY, K.J.: Path integration in mammals. *In: Hippocampus* (2004) 14(2): 80-92

FIELD, D. J.: Relations between the statistics of natural images and the response properties of cortical cells. *In: J Opt Soc Am A* (1987) 4(12): 2379-94

FLANAGAN, J. R. ; WING, A. M.: The role of internal models in motion planning and control: evidence from grip force adjustments during movements of hand-held loads. *In: J Neurosci* (1997) 17(4): 1519-28

FLASH, T. ; SEJNOWSKI, T. J.: Computational approaches to motor control. *In: Curr Opin Neurobiol* (2001) 11(6): 655-62

FLASH, T. ; HOCHNER, B.: Motor primitives in vertebrates and invertebrates. *In: Curr Opin Neurobiol* (2005) 15(6): 660-6

FÖLDIÁK, P.: Learning invariance from transformation sequences. *In: Neural Computation* (1991) 3(2): 194-200

FOSTER, D. J. ; MORRIS, R. G. ; DAYAN, P.: A model of hippocampally dependent navigation, using the temporal difference learning rule. *In: Hippocampus* (2000) 10(1): 1-16

FRANCESCHINI, N.; PICHON, J. M.; BLANES, C.: From insect vision to robot vision. *In: Philosophical Transactions of the Royal Society B* (1992), 337:283-294

FRANZIUS, M. ; SPREKELER, H. ; WISKOTT, L.: Slowness and sparseness lead to place, head-direction, and spatial-view cells. *In: PLoS Comput Biol* (2007) 3(8): e166

FYHN, M. ; MOLDEN, S. ; WITTER, M.P. ; MOSER, E.I. ; MOSER, MB.: Spatial representation in the entorhinal cortex. *In: Science* (2004) 305(5688): 1258-64

GAUSSIER, P. ; REVEL, A. ; BANQUET, J. P. ; BABEAU, V.: From view cells and place cells to cognitive map learning: processing stages of the hippocampal system. *In: Biol Cybern* (2002) 86(1): 15-28

GERSTNER, W. ; ABBOTT, L. F.: Learning navigational maps through potentiation and modulation of hippocampal place cells. *In: J Comput Neurosci* (1997) 4(1): 79-94

GIBSON, J. J.: The Theory of Affordances. *In: Perceiving, Acting, and Knowing.* Eds. Robert Shaw and John Bransford (1977)

GILBERT, P. E. ; KESNER, R. P. ; DECOTEAU, W. E.: Memory for spatial location: role of the hippocampus in mediating spatial pattern separation. *In: J Neurosci* (1998) 18(2): 804-10

GLUCK, M. A. ; MYERS, C. E.: Psychobiological models of hippocampal function in learning and memory. *In: Annu Rev Psychol* (1997) 48:481-514

GRUSH, R.: The emulation theory of representation: motor control, imagery, and perception. *In: Behav Brain Sci* (2004) 27(3): 377-96

HAFTING, T. ; FYHN, M. ; MOLDEN, S. ; MOSER, MB. ; MOSER, E.I.: Microstructure of a spatial map in the entorhinal cortex. *In: Nature* (2005) 436(7052): 801-6

HARNAD, S.: The symbol grounding problem. *In: Physica D* (1990), 42:335-346

HELD, R. ; HEIN, A.:ARNAD, S. : Movement-produced stimulation in the development of visually guided behavior. *In: J Comp Physiol Psychol.* (1963), 56:872-6

HELMHOLTZ, H. von : *Handbuch der physiologischen Optik.* Leipzig: Voss, 1867.

HIPP, J. ; EINHÄUSER, W. ; CONRADT, J. ; KÖNIG, P.: Learning of somatosensory representations for texture discrimination using a temporal coherence principle. *In: Network* (2005) 16(2-3): 223-38

HITZLER, P.; HÖLLDOBLER, S.; SEDA, A.K.: Logic Programs and Connectionist Networks. *In: Journal of Applied Logic* (2004) 2(3):245-272.

HOFFMANN, H.: Unsupervised learning of Visuomotor Asspciations. *PHD thesis, Faculty of Technology, Bielefeld University*, 2005

HOFFMANN, H.: Perception through visuomotor anticipation in a mobile robot. *In: Neural Netw* (2007), 20(1): 22-33

HOK, V. ; LENCK-SANTINI, PP. ; ROUX, S. ; SAVE, E. ; MULLER, R. U. ; POUCET, B. : Goal-related activity in hippocampal place cells. *In: J Neurosci* (2007), 27(3): 472-82

HOMMEL, B. ; MUSSELER, J. ; ASCHERSLEBEN, G. ; PRINZ, W.: The Theory of Event Coding (TEC): a framework for perception and action planning. *In: Behav Brain Sci* (2001), 24(5): 849-78

HONAVAR, V. ; UHR, L.M.: *Artificial intelligence and neural networks: steps toward principled integration.* Boston [u.a.]: Acad. Press, 1994. - 0-12-355055-6

HOPFIELD, J. J.: Neural networks and physical systems with emergent collective computational abilities. *In: Proc Natl Acad Sci U S A* (1982), 79(8): 2554-8

HOSOYA, T. ; BACCUS, S.A. ; MEISTER, M.: Dynamic predictive coding by the retina. *In: Nature* (2005), 436(7047): 71-7

HUBEL, D. H. ; WIESEL, T. N.: Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *In: J Physiol* (1962) 160: 106-54

HUBEL, D. H. ; WIESEL, T. N.: Receptive fields and functional architecture of monkey striate cortex. *In: J Physiol* (1968) 195(1):215-43

HYVARINEN, A. ; HOYER, P.: Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces. *In: Neural Comput* (2000) 12(7): 1705-20

IMBERT, M. ; BUISSERET, P.: Receptive field characteristics and plastic properties of visual cortical cells in kittens reared with or without visual experience. *In: Exp Brain Res* (1975) 22(1):25-36

JÄHNE, B.: *Digitale Bildverarbeitung.* Berlin [u.a.]: 5. Aufl. Springer, 2002. - 3-540-41260-3

JEFFERY, K. J.: Integration of the sensory inputs to place cells: what, where, why, and how? *In: Hippocampus* (2007) 12(9):775-85

KAPLAN, F.; OUDEYER, P.Y.: In search of the neural circuits of intrinsic motivation. *Frontiers in Neuroscience*, (2007) 1(1): 225-236

KARVANEN, J.; CICHOCKI, A.: Measuring sparseness of noisy signals. *In: ICA2003*, (2003)

KAYSER, C. ; KÖRDING, K.P. ; KÖNIG, Peter: Processing of complex stimuli and natural scenes in the visual cortex. *In: Curr Opin Neurobiol* (2004) 14(4):468-73

KLEIN, D.J.; KÖNIG, P.; KÖRDING, K.P.: Sparse spectrotemporal coding of sounds. *In: Eurasip Jasp* (2003) 3:659-667.

KLETTE, Reinhard ; SCHLÜNS, Karsten ; KOSCHAN, Andreas: *Computer vision: three-dimensional data from images.* Singapore [u.a.]: Springer, 1998. - 981-308371-9

KÖNIG, P. ; KRUGER, N.: Symbols as self-emergent entities in an optimization process of feature extraction and predictions. *In: Biol Cybern* (2006) 94(4):325-34

KÖRDING, K. P. ; WOLPERT, D.M.: Bayesian integration in sensorimotor learning. *In: Nature* (2004) 427(6971):244-7

KÖRDING, K.P. ; KAYSER, C. ; EINHÄUSER, W. ; KÖNIG, P.: How are complex cell properties adapted to the statistics of natural stimuli? *In: J Neurophysiol* (2004) 91(1):206-12

KÖRDING, K.P. ; WOLPERT, D.M.: Bayesian decision theory in sensorimotor control. *In: Trends Cogn Sci* (2006) 10(7):319-26

KREIMAN, G. ; KOCH, C. ; FRIED, I.: Imagery neurons in the human brain. *In: Nature* (2000) 408(6810):357-61

KUWANA, Y.; SHIMOYAMA, I.; MIURA, I..: Steering control of mobile robot using insect antenna. *In: IEEE International Conference of Intelligent Robots and Systems* (1995), 2:530-535

LAUGHLIN, S.: A simple coding procedure enhances a neuron's information capacity. *In: Z Naturforsch [C]* (1981) 36(9-10):910-2

LEE, I. ; YOGANARASIMHA, D. ; RAO, G. ; KNIERIM, J.J.: Comparison of population coherence of place cells in hippocampal subfields CA1 and CA3. *In: Nature* (2004) 403(6998):456-9

LEOPOLD, D. A. ; LOGOTHETIS, N. K.: Multistable phenomena: changing views in perception. *In: Trends Cogn Sci* (1999) 3(7):254-264

LEUTGEB, S. ; LEUTGEB, J.K. ; TREVES, A. ; MOSER, MB. ; MOSER, Edvard I.: Distinct ensemble codes in hippocampal areas CA3 and CA1. *In: Science* (2004) 305(5688):1295-8

LEUTGEB, J.K. ; LEUTGEB, S. ; TREVES, A. ; MEYER, R. ; BARNES, C.A. ; MCNAUGHTON, Bruce L. ; MOSER, MB. ; MOSER, E.I.: Progressive transformation of hippocampal neuronal representations in 'morphed' environments. *In: Neuron* (2005) 48(2):345-58

LEWICKI, M. S.; OLSHAUSEN, B. A.: Probabilistic framework for the adaptation and comparison of image codes. *In: J. Opt. Soc. Am. A* (1999) 16(7):1587-601

LEWICKI, M.S.: Efficient coding of natural sounds. *In: Nat Neurosci* (2002) 5(4):356-63

LUNGARELLA, M.; NETTA, G.; PFEIFFER, R.; SANDINI, G.: Developmental robotics: a survey. *Cognitive Science* (2003) 14 (4): 151-190

MARKUS, E. J. ; QIN, Y. L. ; LEONARD, B. ; SKAGGS, W. E. ; MCNAUGHTON, B. L. ; BARNES, C. A.: Interactions between location and task affect the spatial and directional firing of hippocampal neurons. *In: J Neurosci* (1995) 15(11):7079-94

MARR, D.: Simple memory: a theory for archicortex. *In: Philos Trans R Soc Lond B Biol Sci* (1971) 262(841):23-81

MCCALLUM, A. K.: Learning to use selective attention and short-term memory in sequential tasks. *In: From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior* (1996)

MCCALLUM, R. A.: Overcoming incomplete perception with utile distinction memory *In: Proceedings of the Tenth International Conference on Machine Learning* (1993)

MCNAUGHTON, N. ; MORRIS, R. G.: Chlordiazepoxide, an anxiolytic benzodiazepine, impairs place navigation in rats. *In: Behav Brain Res* (1987) 24(1):39-46

MCNAUGHTON, B. L. ; BARNES, C. A. ; GERRARD, J. L. ; GOTHARD, K. ; JUNG, M. W. ; KNIERIM, J. J. ; KUDRIMOTI, H. ; QIN, Y. ; SKAGGS, W. E. ; SUSTER, M. ; WEAVER, K. L.: Deciphering the hippocampal polyglot: the hippocampus as a path integration system. *In: J Exp Biol* (1996) 199(1):173-85

MCNAUGHTON, B. L. ; BATTAGLIA, F. P. ; JENSEN, O. ; MOSER, E. I. ; MOSER, MB.: Path integration and the neural basis of the 'cognitive map'. *In: Nat Rev Neurosci* (2006) 7(8):663-78

MOITA, M. A. P. ; ROSIS, S. ; ZHOU, Y. ; LEDOUX, J. E. ; BLAIR, H. T.: Hippocampal place cells acquire location-specific responses to the conditioned stimulus during auditory fear conditioning. *In: Neuron* (2003) 37(3):485-97

MOORE, A. W.: Variable Resolution Dynamic Programming: Efficiently Learning Action Maps in Multivariate Real-valued State-spaces. *In: Machine Learning: Proceedings of the Eighth International Workshop* (1991)

MORRIS, R. G. ; GARRUD, P. ; RAWLINS, J. N. ; O'KEEFE, J.: Place navigation impaired in rats with hippocampal lesions. *In: Nature* (1982) 297(5868):681-3

MULLER, R. U. ; KUBIE, J. L.: The effects of changes in the environment on the spatial firing of hippocampal complex-spike cells. *In: J Neurosci* (1987) 7(7):1951-68

NAGEL, S. K. ; CARL, C. ; KRINGE, T. ; MARTIN, R. ; KÖNIG, P.: Beyond sensory substitution--learning the sixth sense. *In: J Neural Eng* (2005) 2(4):13-26

NAGEL, S. K.; MÄRTIN, R.; KÖNIG, P.: Fühlen lernen bedeutet Handeln lernen - über das hirnphysiologische Korrelat von Bewusstsein. *In: Deutsche Gesellschaft für Verhaltenstherapie e.V.* (2006) VPP 3/2006

NAITO, E. ; ROLAND, P. E. ; EHRSSON, H. H.: I feel my hand moving: a new role of the primary motor cortex in somatic perception of limb movement. *In: Neuron* (2002) 36(5):979-88

NEWELL, A.: Physical symbol system. *In: Cogntive Science* (1980) 4(2):135-183

NEWELL, A.: *Unified Theory of Cognition.* Havard University Press, Cambridge Ma., 1990

O'KEEFE, J. ; DOSTROVSKY, J.: The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *In: Brain Res* (1971) 34(1):171-5

O'KEEFE, J. ; NADEL, L.: *The hippocampus as a cognitive map.* Oxford: Clarendon Pr., 1978. - 0-19-857206-9;

O'KEEFE, J. ; BURGESS, N.: Geometric determinants of the place fields of hippocampal neurons. *In: Nature* (1996) 381(6581):425-8

O'KEEFE, J. ; BURGESS, N. ; DONNETT, J. G. ; JEFFERY, K. J. ; MAGUIRE, E. A.: Place cells, navigational accuracy, and the human hippocampus. *In: Philos Trans R Soc Lond B Biol Sci* (1998) 353(1373):1333-40

O'REGAN, J. K. ; NOE, A.: A sensorimotor account of vision and visual consciousness. *In: Behav Brain Sci* (2001) 24(5):939-73; discussion 973-1031

O'REILLY, R. C. ; MCCLELLAND, J. L.: Hippocampal conjunctive encoding, storage, and recall: avoiding a trade-off. *In: Hippocampus* (1994) 4(6):661-82

OLSHAUSEN, B. A. ; FIELD, D. J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *In: Nature* (1996) 381(6583):607-9

OLSHAUSEN, B. A. ; FIELD, D. J.: Sparse coding with an overcomplete basis set: a strategy employed by V1? *In: Vision Res* (1997) 37(23):3311-25

OLTON, D.S.; SAMUELSON, R.J.: Remembrance of places passed: Spatial memory in rats. *In: Journal of Experimental Psychology: Animal Behavior Processes*, (1976) 2: 97-116.

OUDEYER, P.-Y.; KAPLAN, F.; HAFNER, V.V.: Intrinsic Motivation Systems for Autonomous Mental Development. *In IEEE Transactions on Evolutionary Computation, Special Issue on Autonomous Mental Development,* 11(2):265-286

PENN, A. A. ; SHATZ, C. J.: Brain waves and brain wiring: the role of endogenous and sensory-driven neural activity in development. *In: Pediatr Res* (1999) 45(4):447-58

PHILIPONA, D. ; O'REGAN, J. K. ; NADAL, J-P: Is there something out there? Inferring space from sensorimotor dependencies. *In: Neural Comput* (2003) 15(9):2029-49

PHILIPONA, D.; O'REGAN, K. J.; NADAL, JP.; COENEN, JO. M. D.: Perception of the Structure of the Physical World using Unknown Multimodal Sensors and Effectors. *In: Advances in Neural Information Processing Systems, 15 2004* (2004)

RAIBERT, M. H.: *Legged Robots that balance.* Cambridge [u.a.]: MIT Press, 1986. - 0-262-18117-7;

RAO, R. P. ; BALLARD, D. H.: Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *In: Nat Neurosci* (1999) 2(1):79-87

REDISH, A. D. ; BATTAGLIA, F. P. ; CHAWLA, M. K. ; EKSTROM, A. D. ; GERRARD, J. L. ; LIPA, P. ; ROSENZWEIG, E. S. ; WORLEY, P. F. ; GUZOWSKI, J. F. ; MCNAUGHTON, B. L. ; BARNES, C. A.: Independence of firing correlates of anatomically proximate hippocampal pyramidal cells. *In: J Neurosci* (2001) 21(5):134

REYNOLDS, S. I.: Adaptive resolution model-free reinforcement learning: Decision boundary partitioning. *In: Proc. 17th International Conf. on Machine Learning* (2000)

ROE, A. W. ; PALLAS, S. L. ; KWON, Y. H. ; SUR, M.: Visual projections routed to the auditory pathway in ferrets: receptive fields of visual neurons in primary auditory cortex. *In: J Neurosci* (1992) 12(9):3651-64

ROLLS, E. T. ; MILWARD, T.: A model of invariant object recognition in the visual system: learning rules, activation functions, lateral inhibition, and information-based performance measures. *In: Neural Comput* (2000) 12(11):2547-72

ROLLS, E. T.: The representation of information about faces in the temporal and frontal lobes. *In: Neuropsychologia* (2007) 45(1):124-43

RUDERMAN, D. L. ; BIALEK, W.: Statistics of natural images: scaling in the woods. *In: Phys. Rev. Lett.* (1994) 73(6):814-17

SAKSIDA, L. M.; RAYMOND, S. M.; TOURETZKY, D. S.: Shaping robot behavior using principles from instrumental conditioning. *In: Robotics and Autonomous Systems* (1997), 22:231-249

SAMSONOVICH, A. ; MCNAUGHTON, B. L.: Path integration and cognitive mapping in a continuous attractor neural network model. *In: J Neurosci* (1997) 17(15):5900-20

SAVE, E. ; CRESSANT, A. ; THINUS-BLANC, C. ; POUCET, B.: Spatial firing of hippocampal place cells in blind rats. *In: J Neurosci* (1998) 18(5):1818-26

Savelli, F. ; Yoganarasimha, D. ; Knierim, James J.: Influence of boundary removal on the spatial representations of the medial entorhinal cortex. *In: Hippocampus* (2008) 18(12):1270-82

SCHAAL, S. ; SCHWEIGHOFER, N.: Computational motor control in humans and robots. *In: Curr Opin Neurobiol* (2005) 15(6):675-82

SCOVILLE, W. B. ; MILNER, B.: Loss of recent memory after bilateral hippocampal lesions. *In: J Neurol Neurosurg Psychiatry* (1957) 20(1):11-21

SEARLE, J.: Minds, brains and programs. *In: Behavioral and Brain Sciences*, (1982), 3:417-424

SHANNON, C. E. ; WEAVER, W.: *The mathematical theory of communication.* Urbana [u.a.]: Univ. of Illinois Press, 1949. - 0-252-72546-8

SHAPIRO, M. L. ; TANILA, H. ; EICHENBAUM, H.: Cues that hippocampal place cells encode: dynamic and hierarchical representation of local and distal stimuli. *In: Hippocampus* (1997) 7(6):624-42

SHI, J.; MALIK, J.: Normalized cuts and image segmentation. *In: IEEE Transactions on Pattern Analysis and Machine Intelligence 2000,* (2000)

SINGH, S.P.; Jaakkola, T.; JORDAN, M. I.: Reinforcement learning with soft state aggregation. *In: Advances in Neural Information Processing Systems 7* (1995)

SIMONCELLI, E. P. ; OLSHAUSEN, B. A.: Natural image statistics and neural representation. *In: Annu Rev Neurosci* (2001) 24:1193-216

SMYTH, D. ; WILLMORE, B. ; BAKER, G. E. ; THOMPSON, I. D. ; TOLHURST, D. J.: The receptive-field organization of simple cells in primary visual cortex of ferrets under natural scene stimulation. *In: J Neurosci* (2003) 23(11):4746-59

SPORNS, O. ; ALEXANDER, W. H.: Neuromodulation and plasticity in an autonomous robot. *In: Neural Netw* (2002) 15(4-6):761-74

SQUIRE, L. R.: Memory and the hippocampus: a synthesis from findings with rats, monkeys, and humans. *In: Psychol Rev* (1992) 99(2):195-231

SRINIVASAN, M. V. ; LAUGHLIN, S. B. ; DUBS, A.: Predictive coding: a fresh view of inhibition in the retina. *In: Proc R Soc Lond B Biol Sci* (1982) 216(1205):427-59

STEELS, L.: Evolving grounded communication for robots. *In: Trends Cogn Sci* (2003) 7(7):308-312

STRINGER, S. M. ; ROLLS, E. T. ; TRAPPENBERG, T. P.: Self-organising continuous attractor networks with multiple activity packets, and the representation of space. *In: Neural Netw* (2004) 17(1):5-27

STROSSLIN, T. ; SHEYNIKHOVICH, D. ; CHAVARRIAGA, R. ; GERSTNER, W.: Robust self-localisation and navigation based on hippocampal place cells. *In: Neural Netw* (2005) 18(9):1125-40

SUMMERFIELD, C. ; EGNER, T. ; GREENE, M. ; KOECHLIN, E. ; MANGELS, J. ; HIRSCH, J.: Predictive codes for forthcoming perception in the frontal cortex. *In: Science* (2006) 314(5803):1311-4

SUR, M. ; GARRAGHTY, P. E. ; ROE, A. W.: Experimentally induced visual projections into auditory thalamus and cortex. *In: Science* (1988) 242(4884):1437-41

SUR, M. ; LEAMEY, C. A.: Development and plasticity of cortical areas and networks. *In: Nat Rev Neurosci* (2001) 2(4):251-62

SUTTON, R. S. ; BARTO, A. G.: *Reinforcement learning: an introduction.* Cambridge, Mass. [u.a.]: MIT Press, 1998. - 0-262-19398-1

TAUBE, J. S. ; MULLER, R. U. ; RANCK, J. B. Jr: Head-direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis. *In: J Neurosci* (1990) 10(2):420-35

TERRAZAS, A. ; KRAUSE, M. ; LIPA, P. ; GOTHARD, K. M. ; BARNES, C. A. ; MCNAUGHTON, B. L.: Self-motion and the hippocampal spatial metric. *In: J Neurosci* (2005) 25(35):8085-96

TISHBY, N.; PEREIRA, F. C.; BIALEK, W.: The information bottleneck method. *In: Proc. of 37th Allerton Conference on Communication and Computation.Monticello, IL.*

TODOROV, E.: Optimality principles in sensorimotor control. *In: Nat Neurosci* (2004) 7(9):907-15

TOLHURST, D.J.; TADMOR, Y.; CHAO, T.: Amplitude spectra of natural images. *In: Ophthalmic Physiol Opt.* (1992) 12:229-32.

TOLMAN, E. C.: Cognitive maps in rats and men. *In: Psychol Rev* (1948) 55(4):189-208

TREVES, A. ; ROLLS, E. T.: Computational constraints suggest the need for two distinct input systems to the hippocampal CA3 network. *In: Hippocampus* (1992) 2(2):189-99

TREVES, A. ; ROLLS, E. T.: *Computational analysis of the role of the hippocampus in memory.* 1994.

TRULLIER, O. ; MEYER, J. A.: Animat navigation using a cognitive graph. *In: Biol Cybern* (2000) 83(3):271-85

ULLMAN, S.: The interpretation of VisualMotion. *MIT, Cambridge, MA* (1979)

VAN DER LOOS, H. ; WOOLSEY, T. A.: Somatosensory cortex: structural alterations following early injury to sense organs. *In: Science* (1973) 179(71):395-8

VAN HATEREN, J. H.: Real and optimal neural images in early vision. *In: Nature* (1992) 360(6399):68-70

VAN HATEREN, J. H. ; RUDERMAN, D. L.: Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. *In: Proc Biol Sci* (1998) 265(1412):2315-20

VAN HATEREN, J. H. ; VAN DER SCHAAF, A.: Independent component filters of natural images compared with simple cells in primary visual cortex. *In: Proc Biol Sci* (1998) 265(1394):359-66

VEREIJKEN, B.; VAN EMMERIK, R.E.A.; WHITING, H.T.A.; NEWELL, K.M.: Free(z)ing degrees of freedom in skill acquisition. *In: Journal of Motor Behavior* (1992) 24:133–142.

VERSCHURE, P. F. M. J. ; VOEGTLIN, T.: A bottom up approach towards the acquisition and expression of sequential representations applied to a behaving real-world device: Distributed Adaptive Control III. *In: Neural Netw* (1998) 11(7-8):1531-1549

VERSCHURE, P. F. M. J. ; ALTHAUS, P.: A real-world rational agent: unifying old and new AI. *In: Cognitive Science* (2003) 27(4):561-590

VINJE, W. E. ; GALLANT, J. L.: Sparse coding and decorrelation in primary visual cortex during natural vision. *In: Science* (2000) 287(5456):1273-6

VINJE, W. E. ; GALLANT, J. L.: Natural stimulation of the nonclassical receptive field increases information transmission efficiency in V1. *In: J Neurosci* (2002) 22(7):2904-15

VOEGTLIN, T. ; VERSCHURE, P. F. M. J.: What can robots tell us about brains? A synthetic approach towards the study of learning and problem solving. *In: Rev Neurosci* (1999) 10(3-4):291-310

VON MELCHNER, L. ; PALLAS, S. L. ; SUR, M.: Visual behaviour mediated by retinal projections directed to the auditory pathway. *In: Nature* (2000) 404(6780):871-6

WALLIS, G. ; ROLLS, E. T.: Invariant face and object recognition in the visual system. *In: Prog Neurobiol* (1997) 51(2):167-94

WEBB, B.: What does robotics offer animal behaviour? *In: Anim Behav* (2000) 60(5):545-558

WEBB, B.: Can robots make good models of biological behaviour? *In: Behav Brain Sci* (2001) 24(6):1033-50; discussion 1050-94

WEHNER, R.; SRINIVASAN, M.V.: Searching behaviour of desert ants, genus *Cataglyphis* (Formicidae, Hymenoptera) *In: J Comp Physiol Psychol* (1981) 142:315-338

WEILLER, D.; LÄER, L.; ENGEL, AK.; KÖNIG, P.: Unsupervised learning of reflexive and action-based affordances to model navigational behavior. *In: KogWis2007*, Saarbrücken (2007)

WIESEL, T. N. ; HUBEL, D. H.: Ordered arrangement of orientation columns in monkeys lacking visual experience. *In: J Comp Neurol* (1974) 158(3):307-18

WILLS, T. J. ; LEVER, C. ; CACUCCI, F. ; BURGESS, N. ; O'KEEFE, J.: Attractor dynamics in the hippocampal representation of the local environment. *In: Science* (2005) 308(5723):873-6

WILLSHAW, D. J. ; VON DER MALSBURG, C.: How patterned neural connections can be set up by self-organization. *In: Proc R Soc Lond B Biol Sci* (1976) 194(1117):431-45

WILSON, M. A. ; MCNAUGHTON, B. L.: Dynamics of the hippocampal ensemble code for space. *In: Science* (1993) 261(5124):1055-8

WISKOTT, L. ; SEJNOWSKI, T. J.: Slow feature analysis: unsupervised learning of invariances. *In: Neural Comput* (2002) 14(4):715-70

WOLPERT, D. M. ; GHAHRAMANI, Z.: Computational principles of movement neuroscience. *In: Nat Neurosci* (2000) 3:1212-7

WYSS, R. ; KÖNIG, P. ; VERSCHURE, P. F. M. J.: Involving the motor system in decision making. *In: Proc Biol Sci* (2004) 271:50-2

WYSS, R. ; KÖNIG, P. ; VERSCHURE, P. F. M. J.: A model of the ventral visual system based on temporal stability and local memory. *In: PLoS Biol* (2006) 4(5):e120

ZHANG, Z.: A Flexible New Technique for Camera Calibration. *In: IEEE Transactions on pattern analysis and machine intelligence* (2000) 22(11): 1330-34

Appendix

Accepted manuscript "WEILLER, D.; LÄER, L.; ENGEL, AK.; KÖNIG, P.: Unsupervised learning of reflexive and action-based affordances to model navigational behavior. *In: KogWis2007*, Saarbrücken (2007)" :

Unsupervised learning of reflexive and action-based affordances to model navigational behavior

DANIEL WEILLER[1] , LEONARD LÄER[1] , ANDREAS K. ENGEL[2] , PETER KÖNIG[1]

[1] Institute of Cognitive Science
Dept. Neurobiopsychology
University of Osnabrück
49069 Osnabrück
Germany

[2] Dept. of Neurophysiology and Pathophysiology
University Medical Center Hamburg-Eppendorf
20246 Hamburg
Germany

Abstract

Here we model animat navigation in a real world environment by using place cell as a sensory representation. The cells' place fields divided the environment into discrete states. The robot learns knowledge of the environment by memorizing the sensory outcome of its motor actions. This was composed of a central process, learning the probability of state-to-state transitions by motor actions and a distal processing routine, learning the extent to which these state-to-state transitions are caused by sensory-driven reflex behavior (obstacle avoidance). Navigational decision making integrates central and distal learned environmental knowledge to select an action that leads to a goal state. Differentiating distal and central processing increases the behavioral accuracy of the selected actions. We claim that the system can easily be expanded to model other behaviors, using alternative definitions of states and actions.

Introduction

Navigation refers to the practice and the skill of animals as well as humans to find their way and to move from one place to another by any means (Wilson and Keil, 1999). The ability of animals to navigate in essentially two-dimensional maze environments have been studied extensively (Olton and Samuelson, 1976; Morris, 1984). Navigation involves cognitive processes like sensory processing, actions execution and decision-making. Here we propose a cognitive model of these processes implemented on a robot faced with the task to navigate in a four-arm-maze environment. To solve this task the robot learns the sensory outcome of its actions, thus acquiring of environmental properties. This knowledge was used to plan and execute actions to solve the navigational task. We claim that the introduced cognitive model is not restricted to a navigational behavior and can easily be generalized to model other behavior.

Navigation can be defined by executing the appropriate action at the right location in an environment to move towards a goal. In the rodents hippocampus O'Keefe (O'Keefe et. al., 1971) found place cells, which encode the position of the animal. These cells fire only when the animal is located in a certain region of the environment, defined as the cell's place field. Although the contribution of these cells to the animal's behavior has still not been fully understood, it is assumed that these cells constitute a cognitive map (O'Keefe and Nadel, 1978) of the environment and are, thus, the bases of navigation. Wyss and coworkers (Wyss et al., 2006) have recently shown that place cells can be understood as an optimally stable representation of the visual input of a behaving robot in a hierarchical network. This implies that unsupervised learning of the sensory input results in a reorganization of the sensory space, spanned by its visual input, which has a spatial meaning. We used this fact by using place cells to locate the robot in its environment. We simplified this task by approximating the firing properties of place cells by Gaussian function and distributed the corresponding place fields in the whole environment. These place fields correspond to the robots internal states and represent the positions between which it is able to differentiate. Hence, in order to enable the cognitive model controlling the robot to perform navigational behavior, we chose place cells as the representation of the environment.

To navigate in the environment the robot first has to learn the sensory outcome of its actions and second to plan its actions according to its knowledge. Learning and planning are both done in its state space, spanned by the place fields. The robot learns local state transitions caused by its action execution. Because the execution of the same actions in a state can result in a transition to different states, the information gained from these local transitions is stored as transition probabilities in a probabilistic directed graph. The robot has also to avoid obstacles. We implemented a reflexive obstacle avoidance behavior controlled by the robots proximity sensors. In case the robot used its reflexive behavior during a transition between two states, we memorized the occurrence of such an event in so called reflex factors. Here the architecture of the cognitive model differentiated between central processing, responsible for state transition memorization and distal processing, responsible for reflex factor learning. The transition probabilities and the reflex factors reflect the environmental properties in relation to the robot's actions. Thus by random action execution, the robot learns an approximation of the environmental affordances (Gibson, 1977), defined the action possibilities afforded by the environment. The robot plans goal directed actions by integrating the information gained by central and distal processing in a local decision-making process. This integration results in a quantitative measure how reliably each executable action leads towards the goal. Overall, the key components of our cognitive model are (i) a high-level representation (place fields) of sensory input space, (ii) the knowledge of environmental properties acquired by active exploration of local state transitions and (iii) a decision-making process driven by this knowledge.

Here we show that using the described cognitive model, a robot can successfully navigate to different goals within a four-arm-maze environment. As expected, the differentiation between central and distal processing reduces the negative effect of the obstacle avoidance behavior on navigational performance. We claim that by redefining the states and actions the introduced model can be expanded to model other behaviors.

Methods

Overview of the architecture

Our cognitive model learns the properties of the environment and plans its action to move towards a goal, based on the state space which is spanned by the spatial representation of place field (state). We divided the four-arm-maze environment (Fig. 1A) into compact discrete states (Fig. 1B) similar to place fields. The architecture of the cognitive model consisted of central and distal components. The central component captures the transition between states, caused by the robots action execution in these states. In contrast the distal component accounts for the usage of distal sensors, like infrared-sensors, facilitating obstacle avoidance during the robots state transition. Here the obstacle avoidance behavior is defined as reflexive behavior. While the central component accounts for any of the robots transition, the distal component constitute only transitions combined with reflexive behavior. Thus the transitions and the transitions combined with reflexive behavior represent the robots locally learned environmental properties according to the robots actions. To navigate to a particular target within the environment, the model chooses during the decision-making process the action that maximally increases the probability of reaching the respective spatial position.

Sensory processing

We chose place cells as a representation of the environment. In a previous study it has been shown how such place cell properties can be acquired by mobile robots using unsupervised learning in a hierarchical network (Wyss et al. 2006). Because here our main purpose is to model behavior we deliberately used predefined place cells to simplify this task. We approximated the firing properties of place cells as a function of the robots position by Gaussian functions (standard deviation: 0.04 m). To cover the whole four-arm-maze environment we randomly distributed 72 of these Gaussian functions. Hence, for each of the robots possible position we obtain the activity of each place cell. A winner-take-all process extracted the robots position in the state space from the activity of the place cells. Accordingly the robot was located in the state (place fields) corresponding to the most activated place cells. (The used distribution of these states is shown in Figure 1B.) In order to determine the robots current state we had to extract its position and calculate the place cells activity using the distributed Gaussian functions as described above. Hence the robot was tracked by a Color Cmos Camera 905C (Analog Camera), which was attached above the environment as shown in Figure 1A. The analog camera signal was digitized by a Hauppauge WinTV Express card. With the help of the camera and the color code attached on top of the robot, its position and orientation were calculated. Thus, the place cell represents a mapping from the position space where the robot is navigating, to the state space of the agent, controlling the robot. The agent uses only positional information provided by this state space (place fields).

Action execution

The robot was able to execute eight different actions in order to restrict the number of transition needed to learn the environmental properties. Each of these actions consisted of a certain orientation followed by a straight movement of the robot. The corresponding orientations were equally spaced from 0 to 325 degrees. As a result of executing such an action in a state (source), the robot will reach a different state (endstate) and thus results in a transition between states. The endstate is defined by the winner-take-all process calculating the current state, being dominated by another place cell. The position within a place field a transition is completed is defined by the place cell's activity not increasing anymore as the robot moves further and thus a local maximum of the activity is reached. The local maximum is defined by the derivative of the robots obtained activity of the place cells being zero. The frequencies of the transitions from source i with action k to endstates j is stored in the experience matrix $EM_{i,j,k.}$.

Reflexes

To prevent the robot hitting one of the mazes boundaries, a reflexive obstacle avoidance behavior was implemented. The proximity sensors (Fig 1C) were used in order to perform this behavior. If the robot had to use its obstacle avoidance behavior during action execution, the system associated the preceding state and action with the occurrence of a reflex event. The frequencies of co-occurrence of the reflexive event and a particular state (j) – action (i) combination is stored in the reflex matrix $RM_{j,i}$.

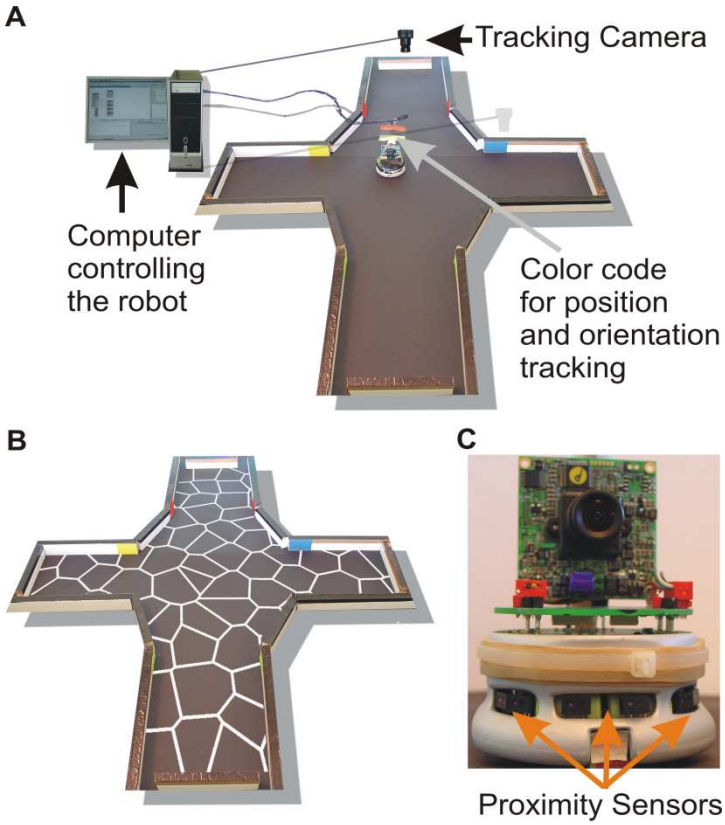

Figure 1: (A) We chose a four-arm-maze environment in order to test the model. A computer controlled the Khepera robot and extracted the robot's position and orientation, using the over head camera and the color code attached on top of the robot. (B) We subdivided the sensory space (position) into states. In the experiments we used this state distribution. The white boundaries assign for the region, one place cells is most activated. (C) Khepera robot used in the navigation experiments.

Decision making

The properties of the environment (boundaries, obstacles, etc.) determine how likely it is that a certain state transition will occur given a chosen action. These state transitions are approximated and learned by the agent as it explores its maze environment and are stored in a transition matrix (Figure 2A). The transition matrix consists of a 2D matrix for each action i TM_i. The row index determines the source j, the state where an action was executed and the row index represents the endstate k of this action. Thus the transition probability defined by source j, endstate k and action i is stored in the transition matrix $TM_{i,j,k}$ shown in Figure 2A. Hence summing of the transition matrix over the endstates k (rows) is normalized to one for each action and sources. For the experiments described below, the robot learned the transition probabilities based on 240 minutes of random exploration.

Next we address the problem of choosing the action with the most desirable outcome to move towards the goal. To accomplish this task an iterative reverse flooding approach was introduced, which

integrates the environmental properties (Fig. 2B). The properties gained from the central component of the architecture are stored in the transition matrix. This matrix consisted of eight 2D matrices TM_i, one for each action, which share similarities with a directed graph. The vertices of this graph correspond to the states, the edges correspond to the transitions and the edges weight to the transition probabilities. This results in 8 directed graphs equivalent to the eight possible actions. In each of the iteration steps of reverse flooding the activation of the state corresponding to the goal states is one. The activation of activated states is propagated through the graph by passing the activity weighted with the corresponding transition probability to the states with transitions to the activated one (reverse direction of the directed edges). Technically spoken the activation is propagated from the endstates to the sources weighted by their transition probability, representing a backward flooding. This process gives rise to 8 different activity values for each state. Thus up to now only the learned environmental properties resulting from the central process were considered during the flooding process. To integrate also the learned properties caused by distal processing we introduced reflex factors. The reflex factor is proportional to the percentage of actions i combined with a reflexive event at source j:

$$rf_{i,j} = 1 - \left(\frac{RM_{i,j}}{\sum_k EM_{i,j,k}} \right) \bullet \frac{5}{6}$$

The weighting factor of 5/6 was introduced to prevent zero activation at an action which is combined only with obstacle avoidance behavior. Thus during each iteration step the eight activations of state j corresponding each to one of the eight actions i were multiplied by the corresponding reflex factor $rf_{j,i}$. After each of the iteration steps the maximum of the eight activations of a state were accounted as the states activation for the next iteration step. This iterative process was continued until the states activity converged. In order to select an action on a state to move towards the goal we considered the eight different incoming state activation values which resulted from the activation propagation of the eight actions. The robot chose the actions which resulted in the highest incoming activation of a state.

Furthermore we introduced a decay factor df which was here 0.9. After each iteration step, the states activation was multiplied by this factor. As more transitions are needed to reach the goal states as more the decay factor is taken into account and thus decreases the states activity. Hence, the decay factor penalized these trajectories to the goal state with more transitions to the goal.

Here the flooding algorithm defined in the last section was implemented with the help of matrices.

$$act_j(0) \begin{cases} 0 & j \neq m \\ 1 & j = m \end{cases}$$

represented the activation at the 0'th activation propagation, where the goal was located at state l.

$$\overrightarrow{act}(t+1) = \max_i((TM_i \cdot \overrightarrow{act}(t)) \cdot rf) \cdot df + \overrightarrow{act}(0)$$

where $\overrightarrow{act}(t)$ is the vector of activation values for the states after t iteration steps. df represents the decay factor.

Robot Setup

To test the model in a real-world environment we used Khepera II robots (K-Team). The robot was equipped with 8 proximity sensors, which emitted infrared light and measured the strength of its reflection, and two wheels, each controlled by one motor (Fig. 1C). For implementation and flexible programming, we used MicroPsi (Bach, 2003; Bach and Vuine, 2003), an Eclipse-based Java programming environment, as an interface to the robot. The agent that controlled the robot's behavior was implemented in this framework. The real-world environment was a four-arm maze with

boundaries built from white wooden pieces (Fig. 1B). Each arm had a width of 0.21 m and a length of 0.28 m. The four-arm maze environment fitted into an area of one meter squared.

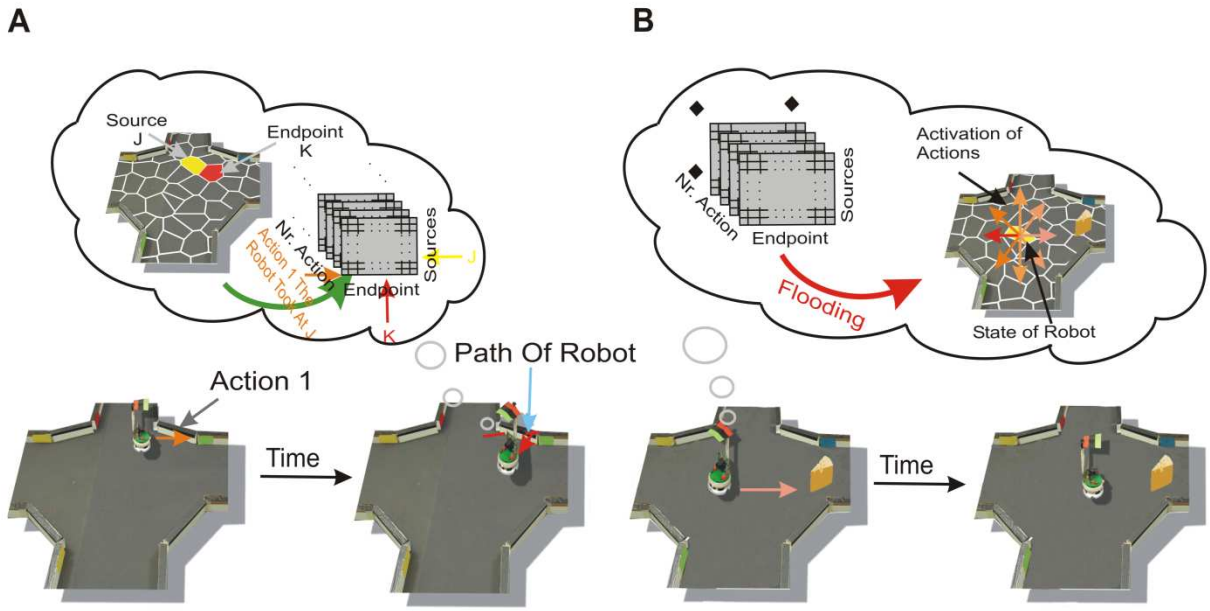

Figure 2**:** (A) Learning of the properties of the environment. The robot is on a certain state, defined here as Source J (yellow labeled) and randomly chooses an action (Action 1). The execution of the action results in another state, defined as endstate K (red labeled). This transition was stored in a three dimensional matrix, called the experience matrix, with the dimensions sources, endpoints and actions. The number of action executions combined with obstacle avoidance from a source was stored separately. (B) The robot moving to a goal (the "cheese" for the artificial "rodent"). His choice is a consequence of the flooding of the transition matrix, resulting in an activation of the different actions, shown as colored arrows. The action with the strongest activation was chosen.

Analysis

As a means of comparison, a simulated robot was implemented using MATLAB (Version 7.0 (R14), Mathworks). The same navigational and experience algorithm was used as described above. The obstacle avoidance behavior was implemented by setting the angle of reflection equal to the angle of incidence to the boundary, with a random scatter of 10 to –10 degrees added.

To compare the navigational behavior and the learned transition of the robot we introduce the geometrical transition matrix. It takes into account only the topographical properties of states in the environment. In order to experience the transition probabilities, based on the topographical properties, we let the simulated robot execute every action on each x/y position within the state given by the resolution of the tracker. Thus the geometrical transition matrix only takes the topographical distributions of states into account. Because the robot chose a new action according to the local maximum of the place cells activity, we weighted an actions transition by the probability of the robot choosing an action at the corresponding cell's activity. The execution of the different actions on each position within a state is due to transition probabilities resulting from an infinite experience time of the robot and thus represents the true underlying transition probabilities.

In order to compare the outcome of the different actions of one state learned by the robot, we measured the correlation coefficient of the transition probabilities of these actions on each state. We correlated the transition probabilities represented by a row vector of the Transition matrix of action i, TM_i, with the same row vector of the Transition matrix of action j TM_j. Before calculating the correlation coefficients between the two vectors we reduced the transition probabilities in the row vector by the average of these transition probabilities. This average was calculated by averaging over

the transition probabilities of the topographical next neighbors. Thus two actions are equivalent when their correlation coefficient is 1.0; they are linearly uncorrelated when the correlation coefficient is 0.0.

To characterize the predictability of an actions' transition to a state we defined a second measure: The predictability of action i in state j is given by the maximum transition probability stored in the row vector j of the Transition Matrix TM_i. This maximum transition probability was reduced by the probability of transferring to one of the connected states by chance.

$$Pr_{i,j} = max_k\left(TM_{i,j,k}\right) - \frac{1}{conn_{i,j}}$$

$Pr_{i,j}$ corresponds to the predictability of action i in state j and $conn_{i,j}$ is the number of states the robot can transfer by executing action i on state j.

In order to evaluate the decision making process we analyzed the activation of each action calculated by the flooding process. We chose the normalized activity as an appropriate measure to characterize the selection of an action during navigating to a goal. This activity is defined as the most activated action on a state normalized by the sum of the incoming activity and the decay factor.

$$NormAct_j = \frac{\overrightarrow{act_j}}{\left(\sum_i((\mathrm{TM_i} \cdot \overrightarrow{\mathrm{act}}(\mathrm{t})) \cdot \mathrm{rf}) \cdot \mathrm{df} + \overrightarrow{\mathrm{act}}(0))_\mathrm{j}\right) \cdot (1 - df)}$$

act_j represents the converged activity of state j after the flooding process. The denominator corresponds to the sum of the activation of a state j over all actions; the activity of state j as well as the sum of activities is given by the converged activity resulting from the flooding process. In order to reduce the dependency of the normalized activation onto the decay factor we multiplied the denominator by this decay factor. Thus normalized activity ranged from 0 to 10.

Results

Here we investigated the robots navigational performance and how the central processes, namely the transition probabilities, as well as the distal processes, defined by the reflex factors, contributed to the decision-making process.

Navigation performance

We investigated the navigation performance of the robot by analyzing its path to a number of different target sites in the environment. In each of the measured trials, the robot was placed on one of five possible starting positions and given one out of four target locations. In order to obtain a comparable measure we normalized the length of the robot's path by the *direct path*. The direct path represented the shortest traversable distance from the robot's starting point to the goal state. Figure 3 shows a path traveled by the robot (yellow line) and the corresponding direct path (light gray line). Overall the robot's median path length across 20 trials was 1.71 with a standard deviation of 0.47. This represents an increase of 71% (± 47%) when compared to the direct path. For all configurations of the start positions and targets, the robot was able to reach the target in a reasonably short amount of time.

This relative increase of the robots path length might have multiple causes: the division of the environment into discrete states (place fields), the robots learned environmental properties transitions and the robots behavior while navigating through the environment. First we investigated the contribution of the discrete states in the lengthening of the robot's path to the targets. To provide a first approximation of this increase, we simulated the robot's behavior using the same navigational algorithm as described in the Methods section. The simulation used the geometrical transition matrix to navigate from the same start positions to the same goal states as the real robot. The transition probabilities of the geometrical transition matrix take only the topographical distribution of states into account (see Method section). Figure 3 shows a path of the simulated robot to a goal (red line). This simulation resulted in a median increase of 19% (±9%) compared to the direct path. Thus, the discrete

states used here to represent the environment did not greatly contribute to the lengthening of the robot's path to a goal.

How can we interpret the robot's navigational behavior? Approximately a quarter of the increase of the robot's path to a goal was caused by the usage of discrete states as a representation of the environment. Another quarter of the lengthening can be explained by the differences between the robots learned and the geometrical properties, stored in the robot's experienced and the geometrical transition matrix (data not shown). Further we analyzed the effect of obstacle avoidance onto the robots navigational performance. The agent engaged its obstacle avoidance behavior in 60% of the trials independent of the particular combination of starting and goal states. Analyzing only the trials in which the agent did not engage obstacle avoidance we obtained a median of 1.36 (±0.23). Thus the largest share of the lengthening of the robot's path compared to the direct path is due the obstacle avoidance behavior. In all trials, the robot was able to find its goal in a reasonably short amount of time, with the main increase in path length arising from the necessity of navigating through the narrow arms of the maze, where obstacle contact occurs most frequently.

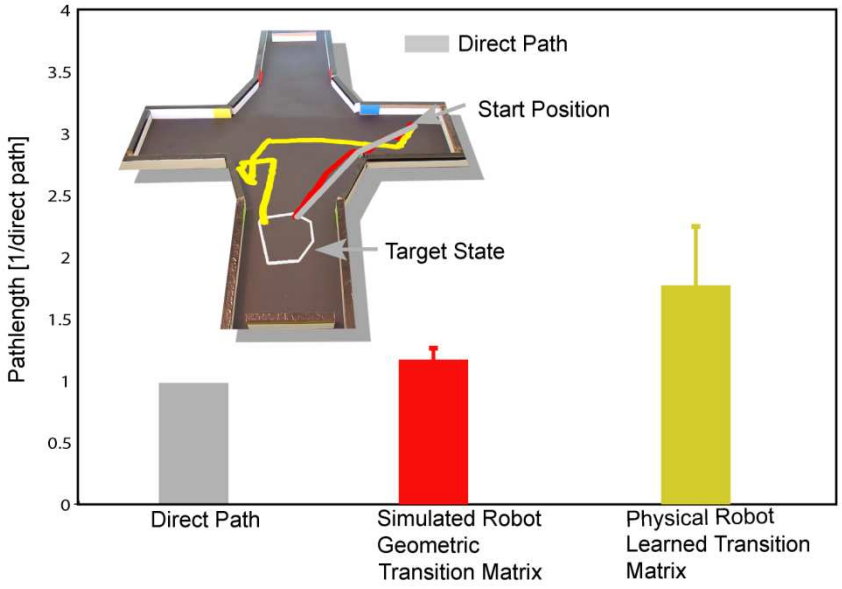

Figure 3**:** Navigational behavior of the robot was investigated by measuring the length of the path to different goals. The direct path, defined as the shortest traversable path from the start point to the goal state (shown as the gray line in the upper part), was used to normalize the length of the robot's path (yellow line) to the goal. The red line corresponds to the length of a path by a simulated robot by taking the topographical distribution of states (geometric transition matrix) into account. The bars represent the median length between different starting and goal states and their standard deviation.

Characteristics of the learned transition matrix

The robot's navigational performance results from the decision-making process. This process is based on the learned transition and the learned reflex factors, representing the learned environmental properties. Here we investigated the characteristics of the robots learned transitions. We investigated first, the differences between the transitions of different actions on a state; second the influence of the used topographical distribution of states on the learned transitions and third the predictability of the state to which one action execution transit to. Fourth we examined the effect of the robots limited time to learn the environmental properties on the learned transition probabilities. For the most part we analyzed the transition matrixes characteristics by comparison to the simulation, based on the geometrical transition matrix (see Method section), representing the transitions based on the used topographical distribution of states. By this comparison we investigate the extent to which the topographical distribution of states gives rise to the investigated characteristics of the transition matrix.

Here we analyzed the similarity between the transitions of different actions, defined as the redundancy of the robot's possible actions on a state, by comparing the transition probabilities associated with these actions. For this purpose we computed correlation coefficients (see Methods and Figure 4A,B) between the transition probabilities of the different actions on each state. Higher correlation coefficients (>0.5) were more frequently observed in the experienced transition matrix (44%) than in the geometrical case (25%), (Figure 4A). Thus the robot's real world action execution resulted in more similar outcomes and thus resulted in a higher redundancy of the actions compared to the geometrical case. Most (93%) of the highly correlated actions in the experienced case were obtained for states at the boundaries of the environment, and so were primarily due to the robot's obstacle avoidance behavior elicited by wall contact. The robot's action execution resulted in more similar transitions compared to the transitions based on the topographical distribution of states.

Next we investigated the influence of the topographical distribution of states on the robot's learned state transitions. Because the used topographical properties of states are fully represented by the geometrical transition matrix (see Method section), we calculated for each state and action the correlation coefficient between the transition probabilities stored in the geometrical and the robot's experienced matrix. Across all actions and states a mean correlation coefficient of 0.56 (±0.52) was obtained. Although these correlation coefficients are low it should be considered that these coefficients are calculated only for neighboring states and thus a conservative estimate. While different actions executed by the robot resulted in similar transitions more often than expected when only the topographical properties of the states are taken into account, the topographical state distribution nevertheless had an influence on the robot's learned transitions.

We then analyzed the predictability of action outcomes. Predictability defines the ability to predict the state to which one action execution makes a transition. In order to evaluate the actions' predictability we introduced predictability values (see Method section), proportional to maximum transition probability of an action. Figure 4D shows the occurrence of predictability values for the experienced and geometric transition matrices. Lower predictability values (<0.3) of the actions occurred more often in the experienced case (37%) compared to the geometric one (13%). Thus in general the robot's actions are equally likely to reach a number of spatially adjacent states. This is due to the actions transition probabilities characterized by a non-sparse probability distribution. Furthermore we investigated the influence of the obstacle avoidance behavior on the action predictability of the experienced transition matrix. Most (84%) of the low predictability values are due to actions for which the robot had to use its obstacle avoidance at least once. Thus obstacle avoidance reduced predictability of the action result. In most cases we obtained a lower predictability of the robot's resultant state, than we would have expected by the topographical distribution of place fields.

Are the differences between the robot's learned and the geometrical properties due to the robot's limited experience time? We generated the transition probabilities of the geometrical transition matrix by simulating the execution of each action on each position within a state (see Method section). Devolving this procedure to the robots learning of the transition probabilities, it has to experience its environment for an infinite time. In contrast, the robot's experienced transition matrix is based on executing each action on each state 11.54 times on average. Here we investigated the influence of the robots limited experience to the mean correlation coefficient between the geometrical and the robots experienced transition probabilities (0.56 ± 0.52). In order to investigate the influence of the robot's limited experience time on the difference between the geometrical and learned matrix, we compared generated geometrical transition matrices to the geometrical transition matrices. The generated geometrical transition matrices were calculated like the geometrical transition matrix; the only difference in the generated case is the number of actions executed on each state restricted to the one of the robots and thus was less than for the geometrical transition matrix. We simulated 300 generated transition matrices. In order to compare these matrices we correlated the transition probabilities for each action and state of the generated matrices with the geometric one. We averaged these correlation coefficients for each generated transition matrix. This yielded a distribution of averaged correlation coefficients with a mean value of 0.86 (± 0.1). Thus, the averaged correlation coefficient of 0.56 (± 0.52) between the geometric and the robots learned transition probabilities were lower than the correlation coefficients between generated and geometrical transition matrix. Thus the difference

between the robot's experienced transition matrix and the geometrical transition matrix is dominated by the robots behavior and not due to limited time the robot experienced the environment.

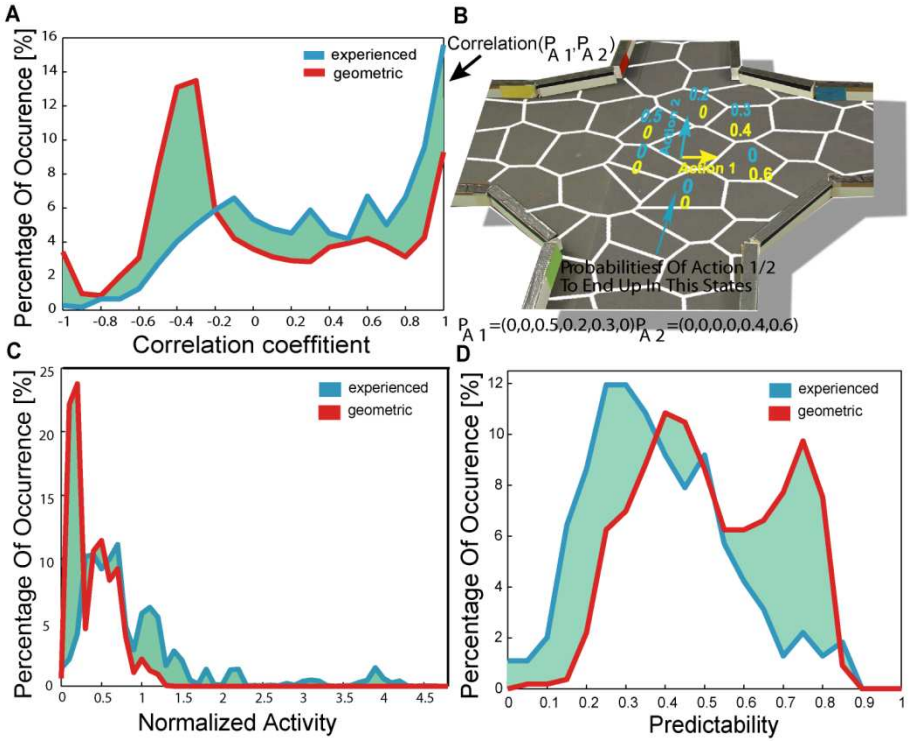

Figure 4: (A) The percentage of occurrence of the different correlation coefficients between the transition probabilities of different actions of a given state. The coefficients for the geometrical and the robot's experienced transition matrix are shown. (B) An example of calculating the correlation coefficients displayed in A. PA1/2 represents the probabilities of the action 1/2. (C) Here the ratio of the most activated action to the sum of all incoming activation (normalized activity (see Method section)) of a state is shown. The higher relfex factors in the experinced case (data not shown) increase the normalized activity compared to the geometric case. (D) Distribution of maximum transition probabilities of all experienced actions and states.

Here we have investigated the properties of the robots learned transition probabilities in the Transition matrix. We obtained a similarity between the transitions of different actions in the robot's experienced compared to the geometrical transition matrix. Also in general a lower predictability of the transition of the robot's actions to a state was obtained compared to the geometrical case. Thus both properties of the robots transition matrix are not fully caused by the topographical distribution of states. The obstacle avoidance behavior gives rise to this lower predictability as well as the similarity of the action results. Neglecting the reflex factors during the decision making process, which navigational behavior would result by only taking the learned transition into account? We would expect that it is not important for the robot to choose a precise action while moving towards a goal, caused by low action predictability as well as the high similarity between the transition probabilities of the different actions. However the transition matrix is influenced by the geometrical distribution of the place fields, while the obstacle avoidance behavior causes a similarity between the actions and a low predictability of an action's resultant state.

Reflexes

Next we investigate the impact of the distal processing on the agent's decision making process, which involves the selection of actions in order to move to a goal. The flooding processes integrates the distal components in the decision making process with help of the reflex factors. After flooding (see

Method section), the agent selects the action most highly activated at the state corresponding to the robot's location. Here we investigate the impact of the reflex factors on this process by analyzing the normalized activation (see Method section). This measure is proportional to the ratio of the highest action's activation to the sum of the other action's activation on a state. Thus a low normalized activation describes a decision making process with the execution of different actions would result in a similar navigational performance. In contrast, high values define a decision making process in which the agent chooses a precise action in order to move to the goal and thus executing different actions than the most activated one would result in different navigational performances. Taking only the transition matrix during the flooding process into account and thus neglecting the reflex factors, based on the properties of the transition probabilities investigated above, we would expect lower normalized activities compared to the geometrical transitions. In contrast, taking the reflexes into account, this normalized activation was higher for the experienced than for the geometric transition matrix (Figure 4C). This implies that the robot chose a precise action in order to move to a goal and thus executing a different action than the highest activated one result in a worse navigational performance. The higher normalized activations for the experienced transition matrix are due to higher reflex factors compared to the geometrical transition matrix (data not shown). Thus taking the reflexes into account reduces the effects of the obstacle avoidance behavior on the learned transitions during the decision-making process and results in a more precise action selection in order to successfully reach a goal.

How do the different components of the algorithm influence the behavior of the robot? Here we analyzed the contribution of the central processes and distal processes to the robots decision-making process. Taking only the central processes, namely the state transitions, for the decision-making into account, different actions executions would result in similar navigational performances, although navigation in the narrow arms required a precise action in order to reduces hits against the walls and thus reduce the path length to goals. In contrast, integrating the distal learned environmental properties, namely reflexes into the decision-making process the robot has to execute one precise action to navigate towards the goal. Thus as we expected, taking the distal processing into account reduces the effects of reflexive behavior and allows the robot to successfully navigate in the environment.

Discussion

Here we have introduced and implemented a model that allows a robot to navigate through an environment. The model learns the environmental properties, in an unsupervised manner by randomly executing the robot's actions possibilities. Because the robots learning process was done in a finite time period, the robots knowledge of its actions possibilities only approximates its environmental affordances. The architecture of this model differentiated between central processing versus distal processing. The distal processing is defined by the state transitions where reflexive behavior of the sensory-driven obstacle avoidance occurred. The central processing is represented by all learned transitions between the states. The reflexive behavior acts upon the robots learned transitions, resulting in uniformly distributed and less predictable actions outcomes than we would have expected by looking at the used topographical distribution of place fields. However, as expected the integration of the information gained by the reflexive and central processing in the decision-making process reduced the impact of sensory-driven behavior on the navigational performance. Consequently the robot was able to successfully navigate in the environment in a short amount of time.

The cognitive model is based on a sensory representation composed of discrete states. In this state space First the robot learned the sensory outcomes of its actions execution, namely the state transition and the reflex factors. Thus, the robot learned the environmental properties with respect to its actions. Based on these results, the robot planned its action in order to move to the goal state in its internal state space. We defined the states such that they are

equivalent to place cells place field, providing a representation of body position within the external space. These place cells can be understood as an optimally stable sensory representation of the visual input given by a robot moving in an environment (Wyss et al., 2006). The unsupervised learning resulted in a reorganization of the sensory space spanned by the robots visual input, leading to a low dimensional representation of the sensory input with a spatially meaning. In order to model other behavior, we have to choose an appropriate organization of the sensory space. On this sensory representation states can be defined, resulting in a state space. Further a definition of actions has to be done, which is adapted to the behavior to be modeled. Corresponding to these actions the sensory outcome in the state space can be learned. Differentiating between distal, namely the transitions influenced by the sensory driven behavior, and the central processes, the state transitions, would result in a better performance of the system to reach a certain goal state. Using a different sensory representation and other definition for the possible actions, different behaviors can be modeled.

Different studies have modeled navigational behavior by using place cells as a representation of the environment. Here we divide the different approaches into two group characterized by the type of learning used: Hebbian learning or reinforcement learning. The first type of learning exploits the fact that while moving in the environment, more than one place cell is active at the rodent's location, caused by the overlapping place fields of the corresponding cells. The Hebbian learning approach takes this fact and applies the biologically motivated principles of LTP and LTD, resulting in a strengthening of the connections between place cells which were active in a certain time interval. These cells and their connections between each other represent a cognitive map (Gerstner and Abott, 1996; Blum and Abott, 1996; Gaussier et al., 2002). Other studies introduced a cell type - goal cells - representing the goal of the navigational task (Burgess et al., 1997; Truellier and Meyer, 2000). The connections between the place and the goal cell encode the place cell's direction to the goal. The strength of connections between these two cell types was also modulated by Hebbian learning. In contrast to our model, the mentioned approaches rely on a global orientation and a metric, measuring the directions and distance to the goal at a given location within the environment. The global orientation used by these studies is defined using the same frame of reference over the whole environment. In contrast, we wanted the robot to learn the topology of the environment and thus did not introduce global variables as orientation or a metric. Furthermore, some of the mentioned studies (Stroesslin et al., 2005; Forster et al, 2000; Gerstner and Abbott, 1997; Burgess et al., 1997; Truellier and Meyer, 2000) used population coding to encode the position or direction to the goal. The population vector approach is based on the assumption of place fields and rodent's orientations having separate topologies. Thus to decode the robot's position or orientation the weighted average of place cells or orientations has to be calculated. This incorporates knowledge of the topology in the decoding scheme and impedes a generalization to other action repertoires. In contrast we defined the actions independently of each other so that the action repertoire can easily be expanded, for example including the action of lifting an object. Other branches of studies (Forster et al., 2000; Aleo and Gerstner, 2000; Stoesslin et al., 2005) used reinforcement learning (Sutton and Barto, 1997) to perform a navigational task. The concepts of Markov Decision Process and value iteration (Sutton and Barto, 1997) are commonalities between reinforcement learning and our approach, while in our model the value iteration was expanded by reflexes. A pure reinforcement learning approach involves learning the properties of the environment by using an explicit reinforcement signal, given by a goal state; while in the presented model these properties are latently learned (Tolman, 1948), resulting in a global strategy for navigation in this environment. In contrast to other studies, here we presented a cognitive model that is able to learn the topology and properties of the

environment in a latent manner and can be expanded to model other behaviors by redefining the meaning of the actions and states.

We introduced a cognitive architecture in order to model animal-like behavior and tested it in a navigational framework. The navigational performance given by this architecture is not constrained to a specific setup because the behaviorally interpreted properties of the environment are self-learned and not predefined. Here we showed that differentiation between central and distal processing routines resulted in a better navigational performance. We argued that this cognitive model can be expanded to model other behavior.

References

Arleo A, Gerstner W. (2000) Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. *Biol Cybern*, 83(3):287-99.

Bach, J. (2003) The MicroPsi Agent Architecture *Proceedings of ICCM-5, International Conference on Cognitive Modeling, Bamberg, Germany* (pp. 15-20)

Bach, J., Vuine, R. (2003) Designing Agents with MicroPsi Node Nets. Proceedings of KI 2003, Annual German Conference on AI. LNAI 2821, Springer, Berlin, Heidelberg. (pp. 164-178)

Blum KI, Abbott LF. (1996) A model of spatial map formation in the hippocampus of the rat. *Neural Comput*. 8(1):85-93.

Burgess N, Donnett JG, Jeffery KJ, O'Keefe J. (1997) Robotic and neuronal simulation of the hippocampus and rat navigation. *Philos Trans R Soc Lond B Biol Sci*. 29;352(1360):1535-43.

Foster DJ, Morris RG, Dayan P. (2000) A model of hippocampally dependent navigation, using the temporal difference learning rule. *Hippocampus. 10(1):1-16*

Gaussier P, Revel A, Banquet JP, Babeau V. (2002) From view cells and place cells to cognitive map learning: processing stages of the hippocampal system. *Biol Cybern*. 86(1):15-28.

Gerstner W, Abbott LF. (1997) Learning navigational maps through potentiation and modulation of hippocampal place cells. *J Comput Neurosci.* 4(1):79-94.

Gibson James J. (1977) The Theory of Affordances. In Perceiving, Acting, and Knowing, *Eds. Robert Shaw and John Bransford*

Morris R (1984) Developments of a water-maze procedure for studying spatial learning in the rat. *J Neurosci Methods* 11 (1): 47-60.

Olten, D.S., & Samuelson, R.J. (1976) Remembrance of places passed: Spatial memory in rats. *Journal of Experimental Psychology: Animal Behavior Processes*, 2, 97-116.

O'Keefe J, Dostrovsky J. (1971) The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Res.* 34(1):171-5.

O'Keefe J. Nadel L. (1978) The Hippocampus as a Cognitive Map. *Oxford University Press*.

Sutton, Richard S.; Andrew G. Barto (1998) Reinforcement Learning: An Introduction. *MIT Press*.

Strosslin T, Sheynikhovich D, Chavarriaga R, Gerstner W. (2005) Robust self-localisation and navigation based on hippocampal place cells. *Neural Netw*. 18(9):1125-40. Epub 2005 Nov 2.

Tolman EC. (1948) Cognitive Maps in Rats and Man. *Psychological Review* 55: 189-208.

Trullier O, Meyer JA. (2000) Animat navigation using a cognitive graph. *Biol Cybern.* 83:271-85.

Wilson R.A., Keil F.C. (1999) The MIT encyclopedia of the cognitive sciences. *MIT Press*

Wyss R, König P, Verschure PF. (2006) A model of the ventral visual system based on temporal stability and local memory. *PLoS Biol.* ;4(5):e120.

Submitted manuscript "Weiller, D.; Läer, L.; Engel, AK.; König, P.: Unsupervised learning of reflexive and action-based affordances to model adaptive navigational behavior." ***Submitted to Autonomous Robots***

Unsupervised learning of reflexive and action-based affordances to model adaptive navigational behavior

DANIEL WEILLER[1] , LEONARD LÄER[1] , ANDREAS K. ENGEL[2] , PETER KÖNIG[1]

[1] Institute of Cognitive Science
Dept. Neurobiopsychology
University of Osnabrück
49069 Osnabrück
Germany

[2] Dept. of Neurophysiology and Pathophysiology
University Medical Center Hamburg-Eppendorf
20246 Hamburg
Germany

Abstract

Here we introduce a cognitive model capable to model a variety of behavioral domains and apply it to a navigational task. We used place cells as sensory representation, such that the cells' place fields divided the environment into discrete states. The robot learns knowledge of the environment by memorizing the sensory outcome of its motor actions. This was composed of a central process, learning the probability of state-to-state transitions by motor actions and a distal processing routine, learning the extent to which these state-to-state transitions are caused by sensory-driven reflex behavior (obstacle avoidance). Navigational decision making integrates central and distal learned environmental knowledge to select an action that leads to a goal state. Differentiating distal and central processing increases the behavioral accuracy of the selected actions and the ability of behavioral adaptation to a changed environment. We propose that the system can canonically be expanded to model other behaviors, using alternative definitions of states and actions.

Introduction

An increasing number of studies model animal behavior using robots. While many of these studies investigate how individual behavioral components, such as sensory processing, contribute to the generation of behavior (Lungarella et al., 2003), most are limited to modeling one particular behavioral domain (Edelman, 2007; Alexander and Sporns, 2002). It is becoming more and more obvious that the flexibility of human behavior is still out of reach of modeling studies (Todorov, 2004; Flash and Sjnowski, 2001). Recently in the neurosciences, different approaches have delineated behavior in a unified theory, independent of any specific behavioral paradigm (Wolpert and Gharamani, 2000; Schaal and Schweighofer, 2005). Here we develop a model based on general principles that we propose to generalize over a broad variety of behavioral domains. For the present study we apply and test it in the domain of a navigational task.

Navigation refers to the practice and skill of animals as well as humans in finding their way and in moving from one place to another by any means (Wilson and Keil, 1999). In other words, a navigational task can easily be described computationally by referring to the position and orientation of the robot as a function of time. Furthermore, the ability of animals to navigate in essentially two-dimensional maze environments has been studied extensively (Olton and Samuelson, 1976; Morris, 1984). To test the cognitive architecture we thus chose an easily determinable navigational task based in the standard environment of a four-arm maze.

To perform a planned behavior, a robot has to predict the sensory outcome of its actions. In order to do so, it is useful to reorganize the high-dimensional sensory input into a low dimensional representation consisting only of behaviorally relevant aspects. Several groups (Wyss et al., 2006; Franzius et al., 2007) have recently shown that the optimally stable representation of a behaving robot's visual input, determined by a hierarchical neural network, is a low dimensional representation equivalent to place cells. The results of learning in these hierarchical networks match the long-standing experimental reports of place cells found in rodent hippocampus (O'Keefe et. al., 1971). These neurons fire only when the animal is located in a certain region of the environment, defining the cell's place field. Although the contribution of these cells to the animal's behavior has still not been fully understood, it is assumed that these cells constitute a cognitive map of the environment (O'Keefe and Nadel, 1978) and are thus the basis of navigation. The work of Wyss and coworkers implies that unsupervised learning of the sensory input results in a reorganization of the sensory space, spanned by its visual input, which has a spatial interpretation. We built on this research by using place cells to locate a robot in its environment. Thus the place fields constitute a discretisation of the navigational state space spanned by the robot's position. They correspond to the robot's internal states and represent the positions it can differentiate. In summary, in order to enable the cognitive model controlling the robot to navigate, we chose place cells as the representation of the environment.

To allow navigation in an environment, we divided the architecture of the agent into central and distal processing. Both processes learn the sensory outcome of the robot's actions in the agent's state space, spanned by the place fields. The central processing component captures the sensory outcomes of the agent's actions considered as state transitions. The agent stores these state transitions as transition probabilities. In contrast, the distal component accounts for the use of distal sensors - here infrared sensors - facilitating obstacle avoidance. Here the obstacle avoidance behavior constitutes a reflexive behavior and its occurrence during a state transition is memorized as a so-called reflex factor. The transition probabilities and the reflex factors reflect the environmental properties in relation to the robot's actions. Thus by random action execution, the robot learns an approximation of the environmental affordances (Gibson, 1977) for navigation, defined as the navigational action possibilities afforded by the environment. The cognitive model plans goal-directed actions by integrating the information gained by central and distal processing into a local decision-making

process. This integration results in a quantitative measure of how reliably each executable action leads towards the goal. In summary, the key components of our cognitive model are (i) a high-level representation (place fields) of sensory input space, (ii) the knowledge of environmental properties acquired by active exploration of local state transitions by means of distal and central processing and (iii) a decision-making process driven by this knowledge.

Here we show that using the described cognitive model, a robot can successfully navigate to different goals within a four-arm-maze environment. As expected, the differentiation between central and distal processing reduces the negative effect of the obstacle-avoidance behavior on navigational performance, and enables the robot to quickly adapt to changes in the environment. We claim that by redefining the states and actions, the introduced model can be expanded to model other types of behavior.

Methods

Overview of the architecture

Our cognitive model allows the robot to experience the environment and navigate to different targets based on a state space represented by the spatial representation of place fields. This state space was obtained by dividing the four-arm-maze environment (Fig. 1A) into compact, discrete states (Fig. 1B), similar to the place fields that can be acquired by unsupervised learning. The central component of the model processes every one of the robot's state transitions, while the distal component deals only with transitions that coincided with reflexive behavior (obstacle avoidance). Together, the transitions induced by the robot's actions and those transitions associated with reflexive behavior represent the environmental properties locally learned by the exploring robot. During each stage of the decision-making process, the model chooses the action that maximally increases the probability of reaching a desired target within the environment, thus allowing the robot to successfully navigate.

Sensory processing

We chose place cells as a representation of the environment. A study by Wyss and colleagues (Wyss et al. 2006) showed that such place cell properties can be acquired by mobile robots by means of unsupervised learning in a hierarchical network. Although it would be possible to replicate this work, our main purpose here is to model behavior, so we deliberately used predefined place cells to simplify the task. We approximated the firing properties of place cells as a function of the robot's position by 2-dimensional Gaussian functions (standard deviation: 0.04 m). To cover the whole four-arm-maze environment we randomly distributed 72 of these Gaussian functions (Figure 1B). For each of the robot's possible positions within the maze, we obtained the activity of each of these place cells. A winner-takes-all process then extracted the robot's position in state space from the population activity of the place cells – the cell that was maximally active thus defined the current state of the agent. In order to calculate the place cell activity, we first needed to extract its position in the environment. The robot was tracked by an Analog Camera (Color Cmos Camera 905C) which was attached

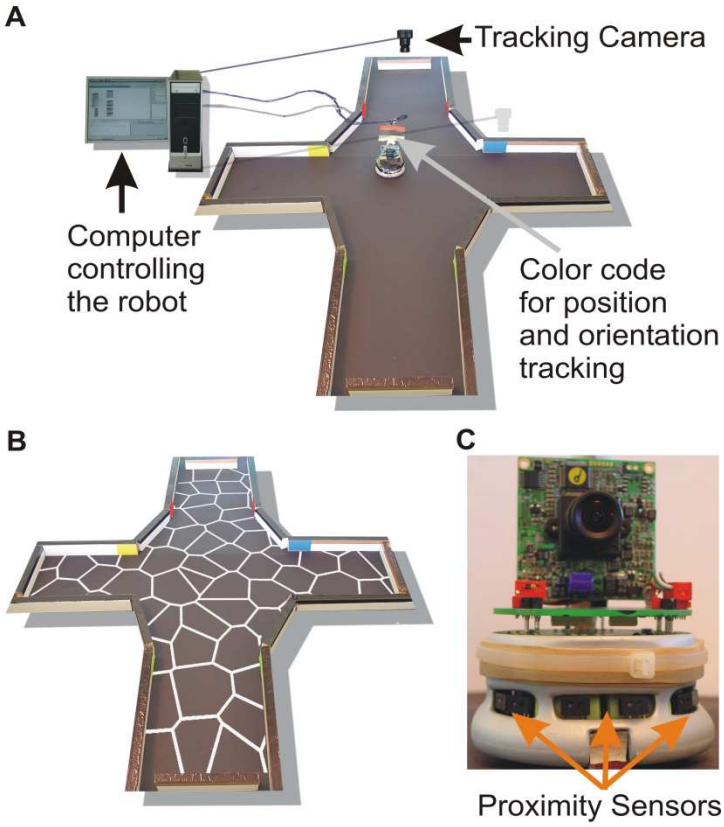

Figure 1: (A) A four-arm-maze environment was chosen to test the model. A Khepra robot was controlled by an agent implemented in the MicroPsi framework, running on a computer. The inputs to the program were the orientation and the position of the robot, as well as the data from the proximity sensors. Part B shows the distributions of states used. The white lines within the environment represent the boundaries of these states. C shows these proximity sensors on the robot. These sensors emit infrared light and measure the reflection. The agent used the activation of the place cells which corresponded to its current position, the orientation of the robot, and the proximity sensors to perform the robot's behavior.

above the environment as shown in Figure 1A. The analog camera signal was digitized by a TV card (Hauppauge WinTV Express). The position and orientation of the robot were then calculated using the camera image and the color code attached on top of the robot. Thus, the population of place cells represents a mapping from the position space in which the robot is navigating, to the state space of the agent controlling the robot. To secure later generalization, no reference is made to the 2-dimensional structure of the environment. The only information used by the agent to infer the robot's position is the activation of each place cell.

Action execution

In order to limit the number of transitions needed to learn the environmental properties to a manageable number, in each state the robot was restricted to executing eight different actions. Each of these actions consisted of a static rotation to a certain orientation followed by the straight-line movement of the robot. The corresponding orientations were equally spaced from 0 to 325 degrees. As a result of executing such an action when in a given state (source), the robot will reach a different state (endstate), with the action thus resulting in a *transition* between states. An endstate is reached when the winner-take-all process calculating the current state returns a new state index. A transition is

defined as complete when a local maximum of the endstate's activity is reached. A local maximum occurs when the derivative of the current state's activity becomes negative. The frequencies of the transitions resulting from action i, executed in source state j and ending in endstate k are stored in the *experience matrix* $EM_{i,j,k.}$.

Distal processing

To prevent the robot hitting one of the maze's boundary walls, a reflexive obstacle avoidance behavior was implemented. The proximity sensors (Fig 1C) were used to perform this behavior. The frequencies of occurrence of the reflexive event characterized by the particular state (i) – action (j) combination is stored in the *reflex matrix* $RM_{i,j}$. Whenever the robot used its obstacle avoidance behavior, the system associated the current state and action with the occurrence of a reflex event and updated the reflex matrix accordingly.

Decision making

The properties of the environment (boundaries, obstacles, etc.) determine how likely it is that a certain state transition will occur given a chosen action. These transition probabilities are approximated and learned by the agent as it explores its maze environment and are stored in a *transition matrix* (Figure 2A). The 3D transition matrix consists of a 2D matrix for each action i TM_i. The row index determines the source state j and the row index represents the endstate k of this action. Thus the transition probability defined by source j, endstate k and action i is stored in the transition matrix $TM_{i,j,k}$ shown in Figure 2A. The sum of the transition matrix over the endstates k (rows) is normalized to one for each action and source and thus represents probabilities. In the experiments described below, the robot learned the transition probabilities are based on 240 minutes of random maze exploration.

Next we address the problem of decision-making – choosing the action that is most likely and quickest to lead to the goal. To accomplish this, an iterative *reverse flooding* approach was introduced, which integrates the environmental properties represented by the transitions and reflex factors (Fig. 2B). The properties learned by the central component of the model are stored in the 8 transition matrices TM_i, and share similarities with a directed graph. The vertices of this graph correspond to the states, the edges correspond to the transitions, and the edge weights to the transition probabilities. This results in 8 directed graphs equivalent to the eight possible actions. In each of the iteration steps of reverse flooding, the activation of the state corresponding to the goal state is set to 1. State activation is propagated through the graph by passing the activity – weighted by the corresponding transition probability – to connected states in the *reverse* direction of the directed edges. Technically speaking, the activation is propagated from endstates to sources, weighted by the transition probability of the action's transfer from the source to the endstate, hence the name reverse flooding. Applying this process to each action's graph gives rise to 8 different activity values for each state. Up to this point, only the learned environmental properties resulting from central processing have been considered during flooding. To further integrate the learned properties caused by distal processing, we introduced reflex factors. The reflex factor is proportional to the percentage of actions i at source j that had an associated reflexive event:

$$rf_{i,j} = 1 - \left(\frac{RM_{i,j}}{\sum_k EM_{i,j,k}} \right) \bullet \frac{5}{6}$$

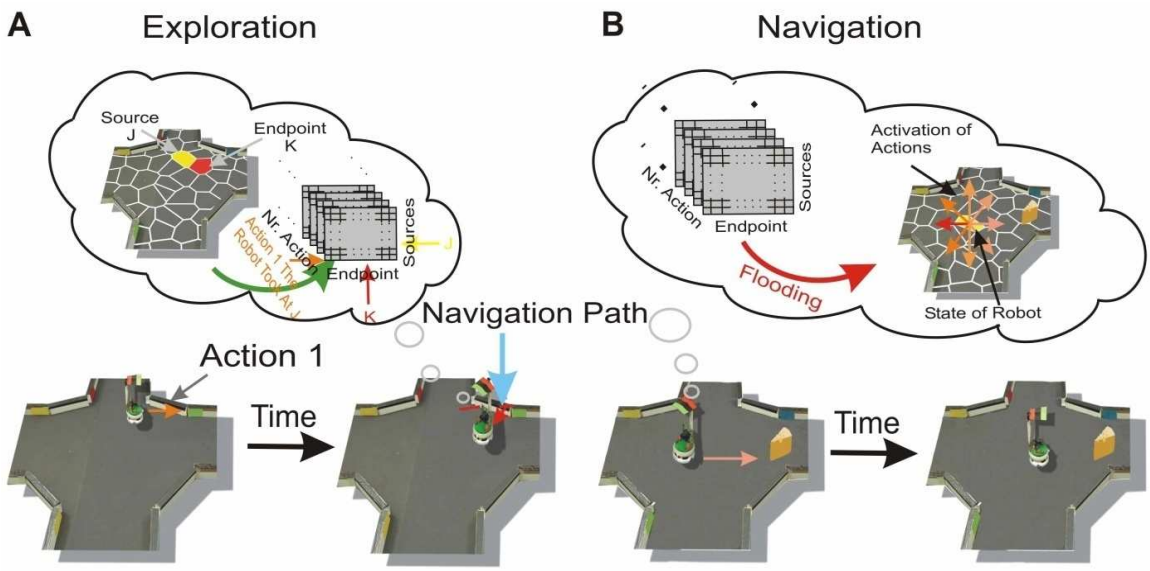

Figure 2: (A) Learning of the properties of the environment. The robot is on a certain state, defined here as Source J (yellow labeled) and randomly chooses an action (Action 1). The execution of the action results in another state, defined as endstate K (red labeled). This transition was stored in a three dimensional matrix, called the experience matrix, with the dimensions sources, endpoints and actions. The number of action executions combined with obstacle avoidance from a source was stored separately. (B) The robot moving to a goal (the "cheese" for the artifical "rodent"). His choice is a consequence of the flooding of the transition matrix, resulting in an activation of the different actions, shown as colored arrows. The action with the strongest activation was chosen.

During each iteration step, the eight activations of state j corresponding each to one of the eight actions i were multiplied by the corresponding reflex factor $rf_{i,j}$. The maximum of the eight activations of a state was used as the state's activation for the next iteration step. This iterative process was continued until the states' activities converged. In order to select the action most likely to move the robot towards the goal, we considered the eight incoming activation values on each state, which resulted from the activation propagation of the eight actions. The robot then chooses the action corresponding to the highest incoming activation of the current state.

For the reflex factors we introduced a weighting factor of 5/6 to prevent a reflex factor of zero in the case of an action which is combined only with obstacle avoidance behavior. Thus, nonzero reflex factors did not neglect the information of the environment gained by the transition probabilities during the flooding process.

Furthermore, we introduced a *decay factor* df, which was here set to 0.9. After each iteration step, the activation of each state was multiplied by this factor. The more transitions that are needed to reach the goal states, the more the decay factor is taken into account and decreases the states' activities. Hence, the decay factor penalized longer trajectories to the goal state.

The flooding algorithm defined above was implemented with the help of matrices.

$$act_j^{(0)} \begin{cases} 0 & j \neq m \\ 1 & j = m \end{cases}$$

represented the activation at the 0'th activation propagation, where the goal was located at state m.

$$\overrightarrow{act}(t+1) = \max_i((\mathrm{TM_i} \cdot \overrightarrow{\mathrm{act}}(\mathrm{t})) \cdot \mathrm{rf}) \cdot \mathrm{df} + \overrightarrow{\mathrm{act}}(0)$$

where $\overrightarrow{act}$(t) is the vector of activation values for the states after t iteration steps. rf represents the reflex factor and df the decay factor.

Robot Setup

To test the model in a real-world environment we used Khepera II robots (K-Team, Lausanne, Switzerland). The robot was equipped with 8 proximity sensors, which emitted infrared light and measured the strength of its reflection, and two wheels, each controlled by one motor (Fig. 1C). For implementation and flexible programming, we used MicroPsi (Bach, 2003; Bach and Vuine, 2003), an Eclipse-based Java programming environment, as an interface to the robot. The agent that controlled the robot's behavior was implemented in this framework. The real-world environment was a four-arm maze with boundaries built from white wooden pieces (Fig. 1B). Each arm had a width of 0.21 m and a length of 0.28 m. The four-arm maze environment fitted into an area of 1 m^2.

Analysis

As a means of comparison, a simulated robot was implemented in MATLAB (Version 7.0 (R14), Mathworks, Natick, MA, USA) using the same algorithms described above. Obstacle avoidance behavior was implemented by setting the angle of reflection equal to the angle of incidence to the boundary, with a random scatter of 10 to –10 degrees added.

To compare the navigational behavior and the transition probabilities learned by the robot, we introduced the *geometrical transition matrix*. This matrix takes into account only the topographical properties of states in the environment and was created by allowing the simulated robot to execute every action on every position within each state, using the resolution of the camera tracking system. Because the real-world robot chose a new action only at a local maximum of its current place cell activity, each transition occurrence in the simulated agent was weighted by the probability of the robot executing an action given the current place cell activity. In an ideal world and given a very long exploration time the real transition matrix is expected to converge to the geometrical transition matrix. In the real world setup, due to the finite robot size, slip and friction and a limited exploration time, the geometrical transition matrix might deviate considerable from the real transition matrix.

Next we evaluated the properties of the experienced and geometrical transition matrices. First we investigated the *similarity of action outcomes* by comparing the corresponding transition probabilities. We correlated the transition probabilities represented by a row vector of the Transition matrix of action i, TM_i, with the same row vector of the Transition matrix of action j TM_j. Before calculating the correlation coefficients between the two vectors we reduced the transition probabilities in the row vector by the average of these transition probabilities to the topographical next neighbors. Thus two actions lead to equivalent outcomes when their correlation coefficient is 1.0; they are linearly uncorrelated when the correlation coefficient is 0.0.

We characterized the predictability of an actions' transition to a state by defining a second measure: The *predictability* of action i in state j is given by the maximum transition probability stored in the row vector j of the Transition Matrix TM_i. This maximum transition probability was reduced by the probability of transferring to one of the connected states by chance.

$$Pr_{i,j} = max_k\left(TM_{i,j,k}\right) - \frac{1}{conn_{i,j}}$$

Here, $Pr_{i,j}$ corresponds to the predictability of action i in state j, and $conn_{i,j}$ is the number of states the robot can reach by executing action i on state j.

In order to evaluate the decision-making process, we analyzed the activation of each action after the flooding process had converged. We chose the *normalized activity* as an appropriate measure

to characterize the strength of selection of an action during navigation to a goal. This activity is defined as the activation of the chosen action for the state, normalized by the sum of all incoming activity and by the decay factor.

$$NormAct_j = \frac{\overrightarrow{act_j}}{\left(\sum_i((\mathrm{TM_i} \cdot \overrightarrow{\mathrm{act}}(\mathrm{t})) \cdot \mathrm{rf}) \cdot \mathrm{df} + \overrightarrow{\mathrm{act}}(0))_\mathrm{j}\right) \cdot (1 - df)}$$

with act_j representing the maximum converged activity of state j after flooding. The denominator corresponds to the sum of all converged activations of state j over all actions. In order to reduce the dependency of the normalized activation on the decay factor, we included the decay factor in the denominator. As a result of its inclusion, the normalized activity ranged from 0 to 10.

Batch and online learning

We investigated the plasticity of the introduced navigational system by examining the robot's navigational adaptation to changes in the environment. The robot's navigational performance was evaluated by measuring its navigational performance to a target. We examined the robot's adaptation process by comparison of the two different types of learning we introduced: *batch* and *online learning*. These approaches differ in the timing of the transition matrix and reflex update and in the way in which the robot explores the environment. Batch learning involves interleaved experience stages of random action execution, during which the existing transition and reflex matrices are updated, and evaluation stages, during which navigation takes place and the transition and reflex matrices are not updated. This is similar to the way the agent experienced the environment as described above. In comparison, online learning involves updating the robot's transition probabilities and reflexes after each action execution, and instead of moving randomly, the decision-making process is always at work.

Results

Here we investigated the robot's navigational performance and how the central processes – namely the transition probabilities – as well as the distal processes defined by the reflex factors, contributed to the decision-making process. We also examined the adaptation of the robot's navigational behavior to changes in the environment.

Navigation Behavior

The navigation performance of the robot was evaluated by repeatedly measuring its path to a number of different target sites in the environment. In each of the 20 trials, the robot was placed on one of five possible starting positions and given one of four target locations. In order to directly compare different start-target combinations, we normalized the length of the robot's path by the *direct path*, which represented the shortest traversable distance from the robot's starting point to the goal state. Figure 3 shows a path traveled by the robot (yellow line) and the corresponding direct path (light gray line). Overall, the robot's median path length across 20 trials was 1.71, with a standard deviation of 0.47. This represents an increase of 71% (± 47%) compared to the direct path length. For all configurations of start positions and targets, the robot was able to reach the target in a reasonably short amount of time.

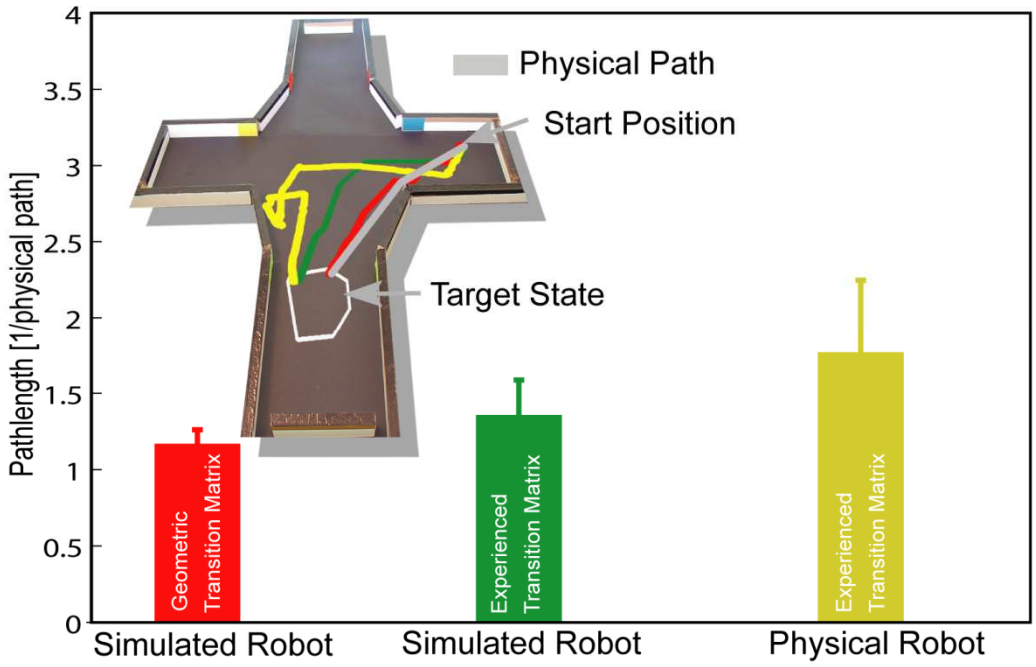

Figure 3: Navigational behavior of the robot was investigated by measuring the length of the path to different goals. The direct path, defined as the shortest traversable path from the start point to the goal state (shown as the gray line in the upper part), was used to normalize the length of the robot's path (yellow line) to the goal. The red line corresponds to the length of a path by a simulated robot by taking the topographical distribution of states (geometric transition matrix) into account. The bars represent the median length between different starting and goal states and their standard deviation.

The increased length of the robot's paths could be a consequence of any of the following: the division of the environment into discrete states (place fields), the environmental properties learned by the robot (transitions and reflex factors), and the robot's behavior while navigating through the environment. Each of these factors was investigated in turn. To provide a first approximation of the increase due to the discretization of the environment, we simulated the robot's behavior using the same navigational algorithm as described in the Methods section. The simulation used the geometrical transition matrix, which takes only the topography of states into account (see Methods), to navigate from the same start positions to the same goal states as the real robot. The red line in Figure 3 shows a sample path of the simulated robot. This simulation resulted in a median increase of 19% (±9%) compared to the direct path. Thus, the introduction of discrete states did not greatly contribute to the lengthening of the robot's path to a goal.

Next we investigated the contribution of the robot's learned environmental properties. To do so, we again used the simulated robot with the same start-target combinations, but this time used the robot's learned transition matrix and reflex factors to perform the task. Figure 3 shows an example of such a simulated path (green line). The median increase in path length was 37% (±23%). As 19 % of the path increase is caused by discrete states, approximately 18 % is due to differences between the geometrical properties of the environment and those properties learned by the robot. Thus, the difference between the geometric and learned transition matrices and reflexes explains a further quarter of the lengthening of the path of the real robot while navigating to a goal. Again, this is a small contribution to the overall increase of the path length.

How can we interpret the robot's navigational behavior? Approximately a quarter of the increase of the robot's path to a goal was caused by the discrete states used to represent the environment. Another quarter of the lengthening can be explained by the differences between the geometrical properties of the environment and those learned by the robot. We also analyzed the effect of obstacle avoidance on the robot's performance. The agent engaged its obstacle avoidance behavior in 60% of the trials, independent of the particular combination of start and goal states. Analyzing only the trials in which the agent did *not* engage obstacle avoidance, we obtained a median path length of 1.36 (±0.23), which is similar to the length measured in simulation with the transition matrix learned by the real-world robot. This is due to operational differences – particularly in obstacle avoidance behavior – between the robot and the simulation (see Method section). Thus the median path length given by the robot's learned transitions represents an approximation of the contribution of obstacle-free navigation. Consequently, the largest share of the lengthening of the robot's path compared to the direct path is due to the obstacle avoidance behavior, which was usually triggered when the robot moved through the narrow arms of the maze. In all configurations of goal states and start positions, the robot was able to find its goal in a reasonably short amount of time, with the main increase in path length arising from obstacle avoidance behavior.

Characteristics of the learned transition matrix

The robot's performance in this navigation task is a direct result of the underlying decision-making process. This process is based on the learned transition and reflex factors, which represent the learned environmental properties. Here we investigated the characteristics of the robot's learned transitions by looking at: i) the differences between the transitions of different actions on a state, ii) the influence of the used topographical distribution of states on the learned transitions of, iii) the number of different states reachable by the different actions, iv) the predictability of the state reachable by a single action execution, and v) the effect of the robot's limited learning time on the learned transition probabilities. For the most part, we analyzed the characteristics of the transition matrices by comparison to the simulation based on the geometrical transition matrix (see Method section), which only takes the used topographical distribution of states into account. This comparison allows us to investigate the extent to which the topographical distribution of states gives rise to the investigated characteristics of the transition matrix.

Here we analyzed the similarity between the transitions of different actions, defined as the redundancy of the robot's possible actions on a state, by comparing the transition probabilities associated with these actions. For this purpose we computed correlation coefficients (see Methods and Figure 4A,B) between the transition probabilities of the different actions on each state. Higher correlation coefficients (>0.5) were more frequently observed in the experienced transition matrix (44%) than in the geometrical case (25%), (Figure 4A). Thus, the robot's real world action execution resulted in more similar outcomes and a higher redundancy of the actions, as compared to the geometrical case. Most (93%) of the highly correlated actions in the experienced case were obtained for states at the boundaries of the environment, and so were primarily due to the obstacle avoidance behavior elicited by wall contact. Overall, the robot's action execution resulted in more similar transitions compared to the transitions based only on the topographical distribution of states.

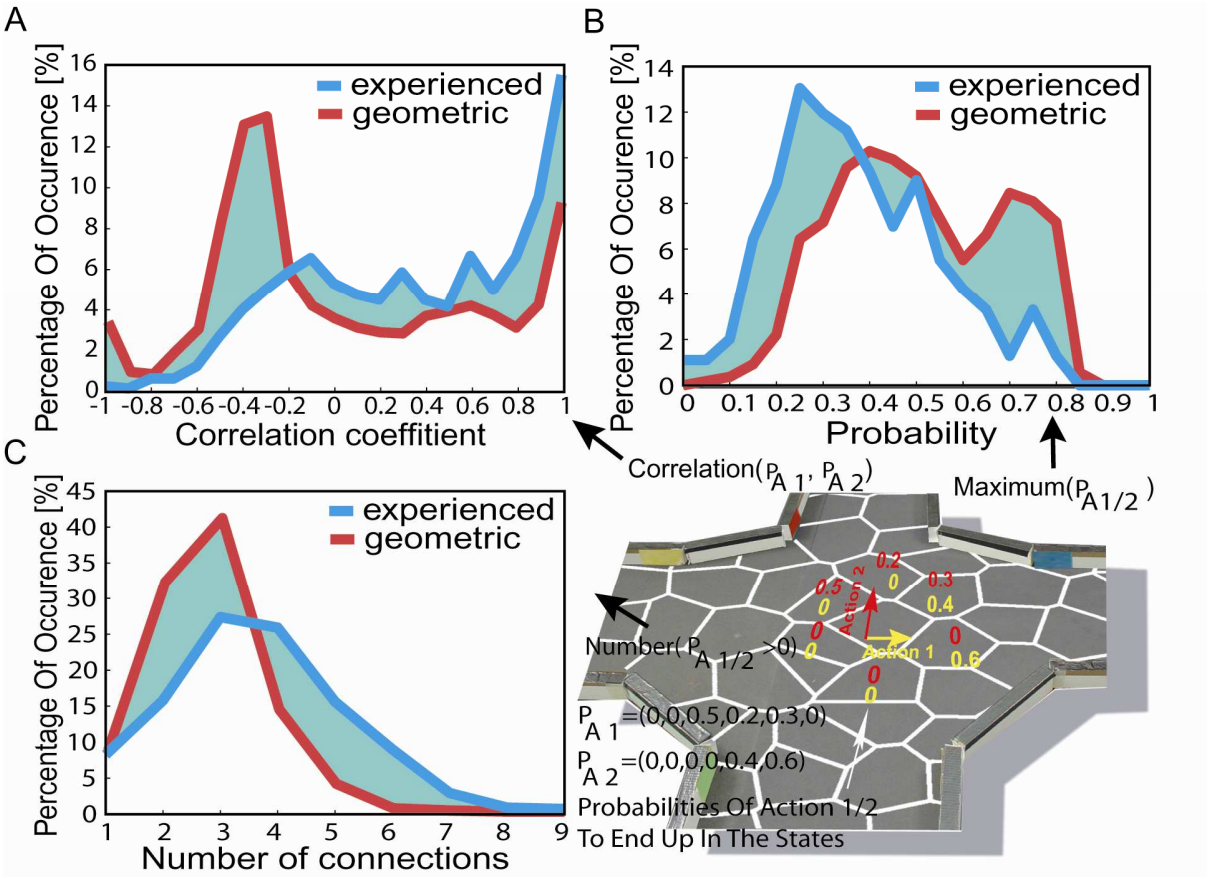

Figure 4: (A) Occurrence of correlation coefficients of the different actions. In order to do so we correlated the transition probabilities to the neighboring states of the actions as shown in the example. (B) The occurrence of the action's highest transition probability, defined as the actions predictability. (C) The actions connectivity, defined by the number of nonzero transition probabilities for each action and state.

Next we investigated the influence of the topographical distribution of states on the robot's learned state transitions. Because the topographical properties of the states we used are fully represented by the geometrical transition matrix (see Method section), we took each state and action and calculated the correlation coefficient between the transition probabilities stored in the geometrical matrix and those stored in the robot's experienced matrix. Across all actions and states, a mean correlation coefficient of 0.56 (±0.52) was obtained. Although these correlation coefficients are low, they should be considered as a conservative estimate of the similarity of action outcomes. This is because the calculation of these coefficients is based only on the transition probabilities to directly neighboring states. However, the transition probabilities to more distant states are mostly zero for all actions, and if these transition probabilities were also included in the correlation calculation, the similarity of different action outcomes would increase. In summary, while the different actions executed by the robot resulted in similar transitions more often than expected when only the topographical properties of the states are taken into account, the topographical state distribution nevertheless had an influence on the robot's learned transitions.

How many different states can possibly be reached by means of a single action? To answer this question we examined the connectivity of the actions, by counting the number of non-zero transition probabilities. Figure 4C shows the occurrence of this connectivity in the geometric and experienced transition matrices. The experienced transition matrix is characterized by a higher

connectivity, with more than half (51%) of all actions showing a connectivity larger than 4 in the experienced case compared to less than a fifth (19%) in the geometric case. In the experienced case the mean connectivity was higher (3.60) than in the geometrical case (2.75). The higher connectivity in the experienced case is due to the obstacle avoidance behavior – in 79% of the experienced actions which led to more than 4 connected states, the robot had to use the obstacle avoidance behavior at least once. More states can be reached by executing a single action in the experienced case.

We then analyzed the predictability of action outcomes. Predictability defines the ability to predict the state that will be reached by a given action execution, and is thus useful for action planning to perceive a certain sensory outcome. In order to evaluate the actions' predictability we introduced predictability values (see Method section) proportional to the maximum transition probability of an action. Although there are alternative ways of measuring predictability (e.g. as the sparseness of transition probabilities), these are similar to the measure used here, because of the normalization of transition probabilities to one. Figure 4D shows the occurrence of predictability values for the experienced and geometric transition matrices. Lower predictability values (<0.3) of the actions occurred more often in the experienced case (37%) compared to the geometric one (13%). Thus in general, the robot's actions are equally likely to reach a number of spatially adjacent states. This is due to the actions' transition probabilities being characterized by a non-sparse probability distribution. Furthermore we investigated the influence of the obstacle avoidance behavior on the action predictability of the experienced transition matrix. Most (84%) of the low predictability values are due to actions for which the robot had to use its obstacle avoidance at least once. In other words, obstacle avoidance reduced the predictability of the action result. In most cases we obtained a lower predictability of the robot's resultant state than we would have expected from the topographical distribution of place fields.

Next we investigated the influence of the obstacle avoidance behavior on the robot's learned transitions. The above investigations of the robot's transitions revealed a reduction in the predictability of the robot's actions, and an increase in the similarity between the robot's action outcomes when compared to the transitions based on the topographical state distribution. These effects on the transitions were due to the robot's engagement of the reflexive obstacle avoidance during these transitions. In other words, the obstacle avoidance behavior acts upon the robot's experience-gathering behavior, thwarting the actions the robot intended to do. Here we investigate the characteristics of the transitions influenced by the reflexive behavior. The obstacle avoidance behavior is guided by proximity sensors, whose activation is highly dependent on the angle of the sensors to an obstacle. These angles can change between different trials, resulting in different sensor activations and thus in different movements of the robot. Thus the outcome of the actions combined with obstacle avoidance has a low reproducibility. A direct result of this low reproducibility is that the transition probability associated with this action will be low, given a high number of experiences. In contrast, a low number of experiences can mean that the transition probabilities of these interrupted actions is high, and will thus have a high influence on the navigational behavior. In order to analyze these effects on the transition probabilities we introduced the notion of a bad connection, defined as a low correlation between the robot's intended actions and the transition that was learned, namely the action's outcome. In order to quantify this relation we calculated the line between the points within a certain state at which the robot chose an action to the point within another state, at which the subsequent action was chosen. This line was compared with the direction of the action the robot intended to take. The executed action was defined as a bad connection if the angle between the line representing the robot's traversed path and the direction of the intended action exceeded 135 degrees. We chose this threshold because the actions with this difference in orientation had a mean correlation of 0.19 (±0.51). Figure 5A shows the mean transition probabilities of these bad connections in the experienced transition matrix as a function of the overall number of gathered experiences. At a low

amount (some hundred) of exploration steps, the average transition probability for these transitions was 0.69. With increasing exploration time, these average probabilities decayed to 0.16. The analyzed connections amounted to 29% of all connections associated with obstacle avoidance behavior. Thus, the influence on the transition matrix of obstacle avoidance resulting in a low correlation between intended and executed action reduces with an increasing number of experiences.

Are the differences between the geometrical properties and those learned by the robot due to the robot's limited experience time? As outlined in the Method section, the geometrical transition matrix was generated by simulating the execution of each action on each position within a state. In order for the real-world robot to learn its environment to this extent simply by executing actions at random, it would have to experience the environment for an infinite time. In contrast, the robot's experienced transition matrix is based on executing each action on each state 11.54 times on average (executing actions on 1.8% over all possible positions within a state; 97% overall action execution was executed only once on a position). Here we investigated the influence of this limited experience on the robot's reduction in action predictability and the increase in the similarity between the outcomes of different executed actions. To do so, we compared the action predictability and action similarity of generated geometrical transition matrices to the geometrical transition matrix. The generated geometrical transition matrices were calculated in the same way as the geometrical transition matrix; however, the number of actions executed on each state was restricted to that of the real-world robot. We simulated 300 generated transition matrices. In order to investigate the influence of finite experience on action predictability, we calculated the action predictability values for each action of the 300 generated transition matrices. We correlated each of these 300 distributions of predictability values with that of the geometrical transition matrix and found a mean correlation of 0.89 (±0.02). In contrast, we obtained a lower similarity (r = 0.48) between the distribution of the predictability values of the robot's experienced transition matrix and the geometrical transition matrix. The same approach was used to correlate the distributions of action similarity values of the generated transition matrices with that of the geometric transition matrix, yielding a high mean correlation of 0.93 (±0.02). In contrast, a low correlation coefficient (0.42) was found between the experienced and geometrical transition matrices. Thus, restricting the amount of experience to that of the robot has a minor effect on the generated geometric transition matrices. Finally, to directly compare the geometrical and generated transition matrices we correlated the transition probabilities for each action and state of the generated matrices with the geometric one. Averaging these correlation coefficients for each generated transition matrix yielded a distribution with a mean value of 0.86 (± 0.01). A lower mean correlation coefficient (0.56) was obtained for the same correlation between the robot's experienced and the geometric transition matrix. The difference between the transition matrix constructed from the robot's experience and the geometrical transition matrix is thus dominated by the behavior of the robot and is not due to limited knowledge of its world.

Here we have investigated the properties of the transition probabilities learned by the robot. In comparison to the transition probabilities given by the topographical distribution of states, we obtained in general a lower predictability of the outcome of the robot's actions, as well as a higher similarity between the outcomes of different actions. These effects are mainly due to the real-world robot's obstacle avoidance. However, despite the differences observed between the geometrical and experienced transition matrix, an influence of the topography of states on the robot's experiences was nonetheless observed. These properties of the transitions are due to the robot's behavior and not to the time-limited experience of the environment. Another influence of the obstacle avoidance behavior on the learned properties of the environment is given by the low correlation between the intended action and the executed action. This influence decreases as the robot increases its experience of the environment. Neglecting the reflex factors (obstacle avoidance behavior) occurring during the decision-making process, which navigational behavior would result by taking only the learned

transitions into account? We would expect that it is not important for the robot to choose a precise action when moving towards a goal, due to the low action predictability as well as the high similarity between the transition probabilities of different actions. Nevertheless, the transition matrix is influenced by the geometrical distribution of the place fields, while the obstacle avoidance behavior causes a similarity between the actions and a low predictability of an action's resultant state.

Decision-making process

The decision-making process involves the selection of actions in order to move to a goal, and integrates the centrally learned properties – namely the transition probabilities – and the distal learned properties – namely the reflex factors. Here we investigate the impact of distal processing on the agent's decision-making process: first in terms of the frequency of obstacle avoidance behavior engaged in by the robot; and second by investigating the influence of the reflex values on the decision-making process. The frequency of obstacle avoidance behavior was quantified as the ratio of transitions combined with reflexive events to the total number of transitions. Figure 4C shows the percentages of occurrence of these reflex values for the geometric and experienced case. Higher values of these ratios occurred more often in the experienced case, with a mean value of 0.42, than in the geometrical case, reflected by a mean of 0.17. This difference in means is caused by operational differences between the robot and the simulation, such as the spatial extension of the robot (see Methods section), which meant that the robot-based agent used the obstacle avoidance behavior more frequently.

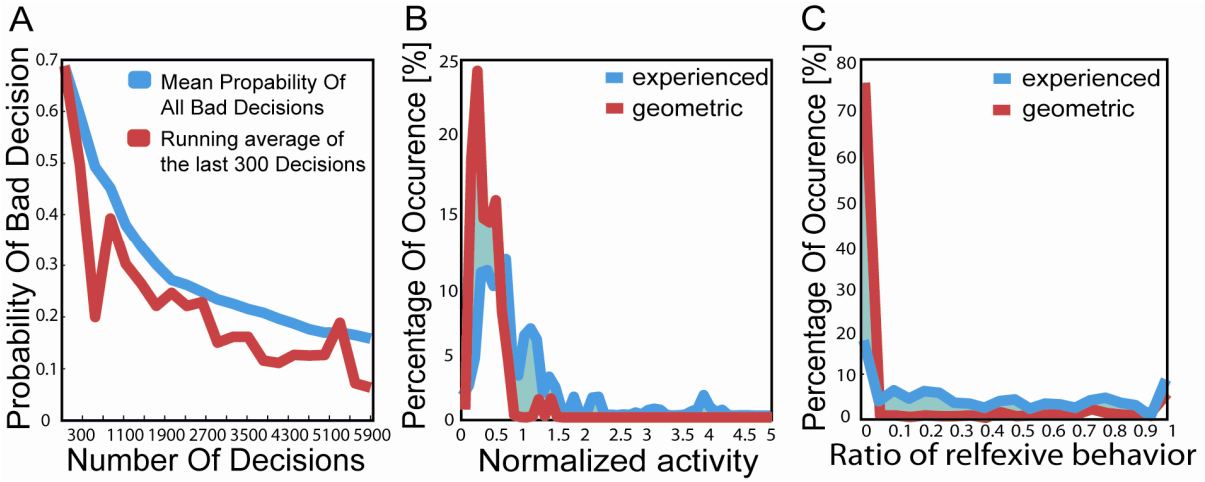

Figure 5: (A) Mean contribution to the transition probabilities of the bad decisions. Bad decision is defined as a low correlation between actions outcome and the direction of the intended action. (B) Occurrence of normalized activity in the decision making process. (C) Ratio of executed actions which resulted in reflexive behavior.

Next we investigated the impact of the reflex factors on the decision-making process by analyzing the normalized activity. After flooding (see Method section), the normalized activity of a state is defined as the ratio of the maximum action activation to the sum of all actions' activations. During the decision-making process, the agent selects the action most highly activated at the robot's current location, which means that a low normalized activity describes a situation where all actions would result in a similar navigational performance. In contrast, high values define a decision-making process in which the agent chooses a precise action in order to move to the goal. In general, this normalized activity was higher for the experienced than for the geometric transition matrix (Figure

4B). This implies that the robot chose a precise action in order to move to a goal, and underwent a stable decision-making process. However, as discussed above, we actually expected a lower normalized activity considering only the transition probabilities. In contrast the lower reflex factors in the experienced case are due to an increase of normalized activities for the experienced transition matrix. Thus taking the reflexes into account reduces the effects of the obstacle avoidance behavior on the decision-making process, and results in a more precise action selection.

How do the different components of the algorithm influence the behavior of the robot? Taking only the central processes, namely the state transitions, for the decision-making into account, different action executions would result in similar navigational performances; although navigation in the narrow arms requires precise actions in order to reduce wall collisions and thus reduce the path length to the goal. Integrating the distal learned environmental properties, namely reflexes, into the decision-making process, the robot now executes one precise action to navigate towards the goal. Thus as we expected, taking the distal processing into account reduces the effects of reflexive behavior and allows the robot to successfully navigate in the environment. As mentioned above, another influence of the reflexive behavior on the transition matrix was a low correlation between intended and executed actions. This influence depended on the extent of the robot's experience in the environment. Taking the reflexes into account reduces the number of experiences needed to neglect this effect on the navigational behavior, as the probabilities combined with obstacle-avoidance behavior were reduced by the reflex factor. Thus the precise selection of an action in the decision-making process and the reduction of the number of experiences needed to navigate in the environment are due to the differentiation between a distal processing represented by the reflex values, and the central processing represented by the transition probabilities between the states, which are both integrated in the decision-making process. Differentiating between reflexive and central processing allows the robot to successfully navigate in the environment.

Learning behavior

Next we analyzed the plasticity of the navigation system by examining the adaptation of the robot's navigational behavior to changes in the environment. In order to do so we inserted an obstacle into the previously learned four-arm-maze environment, as shown in Figure 6A. We implemented and compared two different approaches to allow the robot to adapt to this change: batch and online learning (see Method section).

First we investigated the adaptation process with the help of online learning by analyzing the robot's path passing the added obstacle. In each trial the robot navigated from one start state within one of the three arms to a target site, as shown in Figure 6A. In order to evaluate these trials we calculated the robot's normalized path in a certain area surrounding the wall shown in Figure 6A. Here the normalized path is given by the robot's path in a certain area surrounding the wall, normalized by the direct path, which is the shortest traversable path between the robot's entry and exit point of this area. The lengths of the robot's paths are shown in Figure 6B as a function of trial number. After a few trials (8) the path length of the robot reached values comparable to the navigational performance reported earlier. After 20 trials the selection of actions during decision-making on the different states is stabilized, and thus the changed environmental properties are fully integrated. The variation of the path length in later trials is due to the different start positions of the robot within the different arms (see Figure 6B). Thus, online learning enables the agent to quickly adapt to environmental changes.

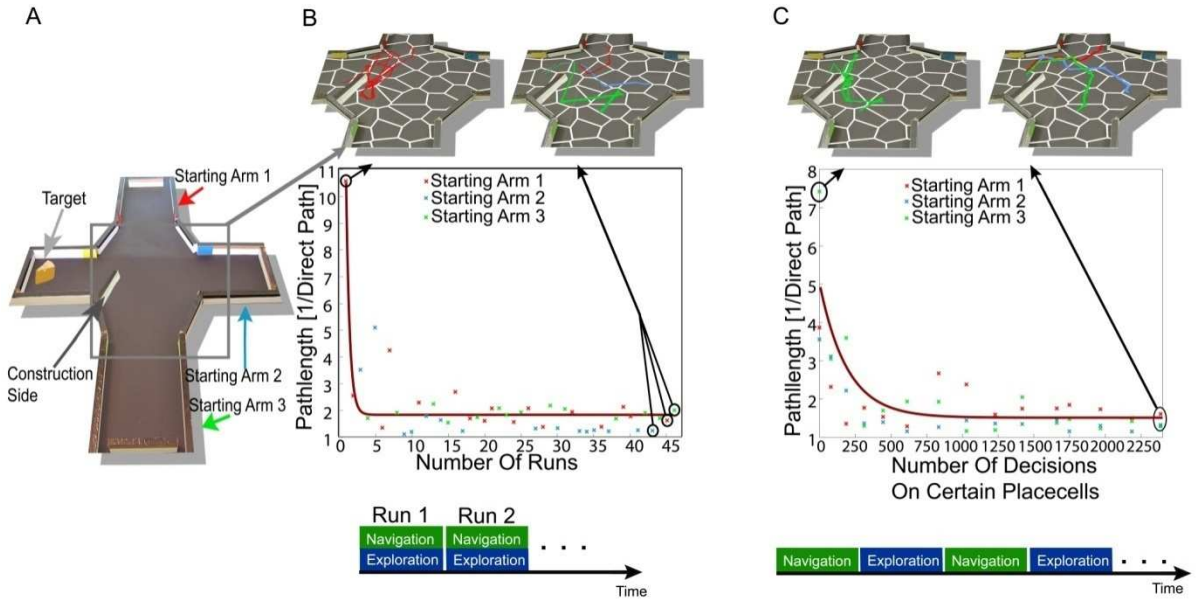

Figure 6: (A) A new obstacle (construction side) is added in an already learned environment. To test the adaptation process we evaluated the robots path through the construction side to the target side starting from three different start states, each located in one of the three different arms. (B) The adaptation process was done with online learning. This type of learning corresponds to an Exploration and Navigation stage at the same time, as shown in the lower part of the figure. The pictures in the upper part show the robot's path in an area around the added obstacle, while the robot navigated to the target. The robot's path length in this area is plotted in the center of this figure. The different colors correspond to the robot's start position in the different arms as shown in A. The brown line corresponds to the best approximation of the path length of all runs by an exponential combined with a constant. (C) The adaptation process using batch learning. After some action executions done in one of the 22 states surrounding the construction side the path of the robot from three different start states to one target side was evaluated. The robots path is shown in the upper part of the figure, for different number of experiences. In the lower part the best decision in order to move to the goal is shown for each state for a different number of experiences.

As already mentioned, the navigational behavior of the system is based on two different environmental properties, stored as transition probabilities and reflexes. In order to analyze their contributions to the environmental adaptation, we compared the decision-making process based on transition probabilities and reflexes to that utilizing the transition probabilities alone. Thus we run the flooding algorithm (as described in the Method section) first using the transition matrix and reflexes, and second utilizing only the transition matrix. In order to analyze the integration of the changed environmental properties into the transition matrix, we compared the action selection based only on the transition matrix before learning and after all trials of online learning. As seen in Figure 7A, we found no difference but on two states between the best action of each state, selected only on the basis of the transition matrix, before (blue arrows) and after online learning (red arrows). Thus experience with the added obstacle is not fully integrated into the transition probabilities. In contrast, the action selection process based on both the transition matrix and the reflexes (yellow arrows) did show integration of the new environmental features. Thus, during online learning, the reflexes are responsible for the integration of the new obstacle into the decision making process.

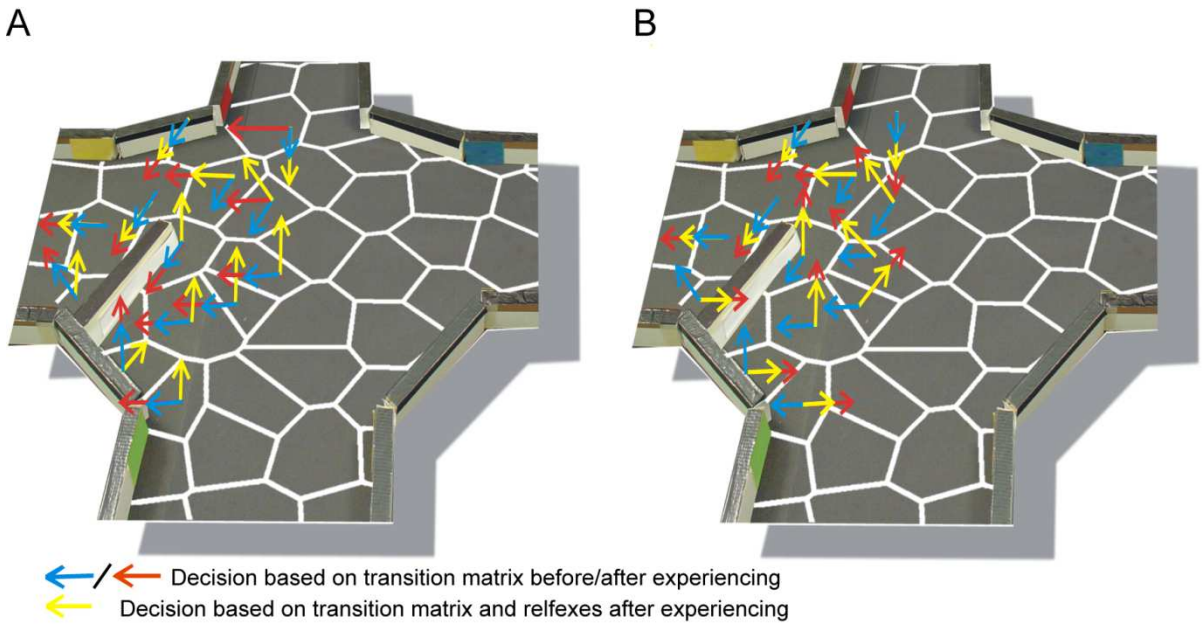

Figure 7: The best decision in order to move to the target side. The different colored arrows correspond to the different conditions. Before the robot experiences the changed environment we executed the decision making process based only on the transition matrix without the reflex factors (Blue arrow). After the adaptation process (online learning: 45 runs/ batch learning: 2500 decisions see figure 6) the decision process was calculated also without reflex factors based on the transition matrix (red arrow). The yellow arrows correspond to the condition of the integrated changes of the environment combined with the transition matrix and the reflexes. (A) represents the online learning while (B) batch learning.

We further investigated this integration of the changed environmental properties into the reflexes rather than the transition probabilities. The adaptation processes of the transition matrix and reflexes are dependent on the amount of actions already executed before the environment was changed (see Method section). As shown previously, without taking the reflexes into account, actions share a similar activation value after the flooding process. In order for the transition matrix to adapt to environmental changes, the connectivity among neighboring states must change. However, we know that connected states share a similar activation due to the similarity and low predictability of the respective action outcomes, which means that any change in transition probability must be reasonably large in order to allow another action to be selected during the decision-making process. Before we added the obstacle to the four-arm-maze, the robot had experienced its environment by executing each action on each state 11.54 times on average. Thus, the robot would have to experience the changed environment for a long time before the change in transition probabilities could trigger an alternative action selection during the decision-making process. In contrast, the reflexive processing acts as a penalty on the action's activation. Thus, the influence of the transition probabilities on the decision-making process depends directly on the activation of the neighboring states. In contrast to the influence of the reflex factors, which depends on the sum of the incoming activation for each action and thus conveniently require less experiences to integrate any environmental changes. As soon as the reflexes have adapted to the changes in the environment, the action which leads to a reflexive behavior is no longer executed during the online learning process. As a result, the adaptation process stops. In summary, during online learning it is the reflexive processing that enables a fast integration of the environmental changes.

Next we investigated the adaptation process involved in batch learning and compared it to online learning. Each of the experience stages was specified by the number of action execution done on each of the 22 states surrounding the added obstacle. In order to evaluate the robots navigational performance, the robot navigated in each navigation stage two times from three different start positions within one arm to the target site. The average normalized path within a certain area around the added obstacle, for each start position is shown in Figure 6C. After an experience stage containing around 500 experiences, the path length of the robot reached a length comparable to the robot's best navigational performance reported earlier. A slight increase in the path length can be seen after 800 experiences. This is due to some new learned features of the environment. As the obstacle is located in the middle of a state, actions are executed which results in a reflexive behavior on one side of the obstacle and on the other side not. Thus in this case batch learning could result in some instability in the decision making process, according to some conflicting experiences learned in the environment. After 1500 randomly executed actions the navigational performance does not change much anymore and in general the best actions in order move to the target did not change anymore. Thus after a short amount of time the changed environmental features are integrated in the navigational performance by batch learning. Here, the robot needed more time to experience the environment compared to online learning. This is due to the difference types of learning, as online learning integrates only the environmental features in order to move to the goal while batch learning is latent learning and thus could integrate any changed features. However after a short amount of time online and batch learning integrates the environmental changes in their navigational behavior.

Also here we analyzed the contribution of the transition probabilities and the reflex factors to the navigational adaptation. As done for online learning we analyzed the decision-making process by comparison of the decision-making process based on the transition matrix and reflexes with the one based on only the transition matrix. We concluded from figure 7B that also the transition probabilities adapted to the changes in the environment. Thus in contrast to online learning, during batch learning the transition probabilities are able to integrate the changed environmental features.

Differentiating between reflexive and central processing allows the robot to successfully navigate in the environment. This differentiation also results in a fast integration of the environmental changes and thus navigational adaptation to the changes in the environment. The presented architecture is able to successfully model navigational behavior and keeps its plasticity in an already learned environment.

Discussion

We have introduced a cognitive model capable of generalizing over a broad variety of behavioral domains, and applied it to a navigational task. Here, behavior was modeled as state transitions in the state space spanned by place cells. Furthermore, the architecture of this model differentiated between central processing and distal processing. Distal processing is defined by the state transitions where the reflexive behavior of the sensory-driven obstacle avoidance was triggered. Central processing acts on all learned transitions between states. The reflexive behavior acts upon the robot's learned transitions, resulting in uniformly distributed and less predictable action outcomes than expected from inspection of the topographical distribution of place fields used. However, as expected, the integration of the information gained by reflexive and central processing in the decision-making process reduced the impact of sensory-driven obstacle-avoidance behavior on the navigational performance. In addition the introduced model quickly adapts to changes in the environment. Consequently, the robot was able to successfully navigate in the environment after only a short amount of time.

Here we used 8 different discrete actions in order to limit the robot's experience time. The outcome of these different actions resulted in redundancies, which would increase by increasing the

number of actions. Consequently the robot would not gain more information about the environment by more action possibilities. In addition using discrete actions allows the cognitive architecture to be easily expandable. Without a change of concepts it might be applied to a robotic arm, e.g. lifting objects. Admittedly, in a very high dimensional state space new problems due to very sparse data arise. In this cognitive architecture the limitation on modeling different behaviors is given by the robot's experience time, which increases with the number of states and actions and thus with the complexity of the behavior to be modeled.

Different studies have modeled navigational behavior by using place cells as a representation of the environment. These different approaches can be characterized by the type of learning used: Hebbian learning or reinforcement learning. The first type of learning exploits the fact that while moving in the environment, more than one place cell is active at the rodent's location, caused by the overlapping place fields of the corresponding cells. This allows the application of the biologically motivated principles of LTP and LTD, resulting in a strengthening of the connections between place cells which were active in a certain time interval. These cells and their connections between each other represent a cognitive map (Gerstner and Abott, 1996; Blum and Abott, 1996; Gaussier et al., 2002). Other studies introduced a cell type - goal cells - representing the goal of the navigational task (Burgess et al., 1997; Truellier and Meyer, 2000). The connections between the current place and the goal cell encode the place cell's direction to the goal. The strength of connections between these two cell types was also modulated by Hebbian learning. In contrast to our model, the mentioned approaches rely on a global orientation and a metric, measuring the direction and distance to the goal from a given location within the environment. The global orientation used by these studies is defined using the same frame of reference over the whole environment. In contrast, we wanted the robot to learn the topology of the environment and thus did not introduce such global variables as orientation or a metric. Furthermore, some of the mentioned studies (Stroesslin et al., 2005; Forster et al, 2000; Gerstner and Abbott, 1997; Burgess et al., 1997; Truellier and Meyer, 2000) used population coding to encode the position or direction to the goal. The population vector approach is based on the assumption that place fields and rodent's orientations have separate topologies. Thus to decode the robot's position or orientation, the weighted average of place cells or orientations has to be calculated. This incorporates knowledge of the topology in the decoding scheme and impedes a generalization to other action repertoires. In contrast, we defined the actions independently of each other so that the action repertoire can easily be expanded, for example including the action of lifting an object. Other branches of studies (Forster et al., 2000; Aleo and Gerstner, 2000; Stoesslin et al., 2005) used reinforcement learning (Sutton and Barto, 1997) to perform a navigational task. The concepts of Markov Decision Processes and value iteration (Sutton and Barto, 1997) are commonalities between reinforcement learning and our approach, while in our model, value iteration was expanded by reflexes. A pure reinforcement learning approach involves learning the properties of the environment by using an explicit reinforcement signal, given by a goal state; in the presented model these properties are latently learned (Tolman, 1948), resulting in a global strategy for navigation in this environment. In contrast to other studies, here we have presented a cognitive model that is able to learn the topology and properties of the environment in a latent manner and can additionally be expanded to model other behaviors by redefining the meaning of the actions and states.

The introduced cognitive model is based on a sensory representation composed of discrete states. In this state space the robot first learned the sensory outcomes of its action's execution, namely the state transition and the reflex factors. Thus, the robot learned the environmental properties in an unsupervised fashion with respect to its actions. Based on these results, the robot planned its action in order to move to the goal state. We defined the states such that they are equivalent to the place fields of place cells, providing a representation of body position within the external space. These place cells can be understood as an optimally stable sensory representation of the visual input to a robot moving

in an environment (Wyss et al., 2006). The unsupervised learning resulted in a reorganization of the sensory space spanned by the robot's visual input, leading to a low-dimensional representation of the sensory input with a spatial interpretation. In order to determine the robot's current state, we used a winner-takes-all process over all place cells' current activations. This process results in a discrete division of the robot's navigational space, spanned by the robot's possible positions. Since this navigational space completely determines the navigational task, a discrete division of this space corresponds to a discrete division of any sensory space relevant for navigation (e.g. the sensory space spanned by the proximity sensors, which measure distances to objects). In addition, differentiating between distal processes – namely the transitions influenced by the sensory-driven behavior – and central processes – the state transitions – should result in a better performance of the system. In order to extend our model to different behaviors, we need only to divide the relevant sensory space into discrete states and implement the actions that define the behavior to be modeled. By means of executing these actions, the sensory outcome in the state space can be learned.

We have introduced a cognitive architecture for the modeling of animal-like behavior, and plan to use this model to obtain further insights into the general principles of behavior, such as action planning. The model is based on a discrete division of sensory space into states, and here the state space was obtained by randomly distributing place cells in the environment. The topographical distribution of these states was shown to influence the transition probabilities, and thus to affect the robot's behavior. This raises the important question of how these states should be organized to allow the robot to behave optimally. In other words, how can we optimally reorganize the sensory space, given only the sensory input and the robot's action repertoire? This corresponds to a reorganization of the somatosensory space, which is recently highly discussed topic (Oudeyer et al., 2007; Kaplan and Oudeyer, 2007; McCallum, 1993; Moore, 1991; Singh et al., 1995). As our model can generalize over different behavioral domains, we claim that it can be employed to gain insights into different behavioral principles such as those mentioned here.

References

Alexander H.W. and Sporns O. (2002): An Embodied Model of Learning, Plasticity and Reward. *Adaptive Behaviour* 10: 143-159

Arleo A, Gerstner W. (2000) Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. *Biol Cybern,* 83(3):287-99.

Bach, J. (2003) The MicroPsi Agent Architecture *Proceedings of ICCM-5, International Conference on Cognitive Modeling, Bamberg, Germany* (pp. 15-20)

Bach, J., Vuine, R. (2003) Designing Agents with MicroPsi Node Nets. Proceedings of KI 2003, Annual German Conference on AI. LNAI 2821, Springer, Berlin, Heidelberg. (pp. 164-178)

Blum KI, Abbott LF. (1996) A model of spatial map formation in the hippocampus of the rat. *Neural Comput*. 8(1):85-93.

Burgess N, Donnett JG, Jeffery KJ, O'Keefe J. (1997) Robotic and neuronal simulation of the hippocampus and rat navigation. *Philos Trans R Soc Lond B Biol Sci*. 29;352(1360):1535-43.

Edleman G.M. (2007). Learning in and from Brain-Based Devices. *Science* 318:1103-1105

Flash T. and Sejnowski TJ. (2001) Computational approaches to motor control. Curr Opin Neurobiol. Dec;11(6):655-62.

Foster DJ, Morris RG, Dayan P. (2000) A model of hippocampally dependent navigation, using the temporal difference learning rule. *Hippocampus. 10(1):1-16*

Franzius, M., Sprekeler, H. & Wiskott, L. (2007) Slowness leads to place, head-direction, and spatial-view cells. *PLoS Computational Biology 3(8):e166.*

Gaussier P, Revel A, Banquet JP, Babeau V. (2002) From view cells and place cells to cognitive map learning: processing stages of the hippocampal system. *Biol Cybern*. 86(1):15-28.

Gerstner W, Abbott LF. (1997) Learning navigational maps through potentiation and modulation of hippocampal place cells. *J Comput Neurosci.* 4(1):79-94.

Gibson J. J. (1977) The Theory of Affordances. In Perceiving, Acting, and Knowing, *Eds. Robert Shaw and John Bransford*

Kaplan, F. and Oudeyer, P-Y (2007) In search of the neural circuits of intrinsic motivation. *Frontiers in Neuroscience*, 1 (1), 225-236

Lungarella M., Netta G., Pfeiffer R., Sandini G. (2003): Developmental robotics: a survey. *Cognitive Science* 14 (4): 151-190

McCallum R. A. (1993): Overcoming incomplete perception with utile distinction memory. *The proceeding of the tenth international machine learning conference.* 1993

Moore A. W. (1991): Variable resolution dynamic programming: Efficiently learning action maps in Multivariate real-valued state-spaces. *Proceedings of the eight international workshop of machine learning. 1991: 333-337.*

Morris R (1984) Developments of a water-maze procedure for studying spatial learning in the rat. J Neurosci Methods 11 (1): 47-60.

Olten, D.S., & Samuelson, R.J. (1976) Remembrance of places passed: Spatial memory in rats. *Journal of Experimental Psychology: Animal Behavior Processes,* 2, 97-116.

Oudeyer P.-Y., Kaplan F., Hafner V.V. (2007), Intrinsic Motivation Systems for Autonomous Mental Development, *IEEE Transactions on Evolutionary Computation, Special Issue on Autonomous Mental Development,* Volume: 11, Issue: 2, pp. 265-286

O'Keefe J, Dostrovsky J. (1971) The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Res.* 34(1):171-5.

O'Keefe J. Nadel L. (1978) The Hippocampus as a Cognitive Map. *Oxford University Press.*

Schaal S. and Schweighofer N. (2005): Computational motor control in humans and robots. Current Opinion in Neurobiology 15:675-682

Singh S. P., Jaakkola T., Jordan M. I. (1995): Reinforcement learning with soft state aggregation. Advances in Neural Information Processing Systems 7. MIT Press

Sutton, Richard S., Andrew G. Barto *(1998)* Reinforcement Learning: An Introduction. *MIT Press.*

Strosslin T, Sheynikhovich D, Chavarriaga R, Gerstner W. (2005) Robust self-localisation and navigation based on hippocampal place cells. *Neural Netw.* 18(9):1125-40. Epub 2005 Nov 2.

Todorov E. (2004) Optimality principles in sensorimotor control. Nat Neurosci. Sep;7(9):907-15.

Tolman EC. (1948) Cognitive Maps in Rats and Man. Psychological Review 55: 189-208.

Trullier O, Meyer JA. (2000) Animat navigation using a cognitive graph. *Biol Cybern.* 83:271-85.

Wilson R.A., Keil F.C. (1999) The MIT encyclopedia of the cognitive sciences. *MIT Press*

Wolpert D.M. and Ghahramani Z. (2000): Conputational principles of movement in neuroscience. Nat Neurosci. 2000 Nov;3 Suppl:1212-7.

Wyss R, König P, Verschure PF. (2006) A model of the ventral visual system based on temporal stability and local memory. *PLoS Biol.* ;4(5):e120.

Acknowledgements

I would like to thank my supervisors Peter König and Andreas Engel, without them this work would not have been possible. Furthermore, I would like to thank all coauthors who contributed to the presented work: Sven Dähne, Leonhard Läer and Robert Märtin. I especially thank Frank Schuman for nice discussions. Further, I would like to thank Kathrin Obst, Marlene Weiss and Daniel Bracko, for their patience and their company in the moments of despair. Further I would like to thank Cliodhna Quigley for improving my capability to communicate with the world and the improvement of the language in this thesis.

Affirmation

I hereby confirm that I wrote this thesis independently and that I have not made use of resources other than those indicated. Further, this thesis was neither published in Germany nor abroad, except the parts of this thesis indicated above, and has not been used to fulfill any other examination requirements.

Osnabrück, February 16th 2009