

DISS. ETH NO. 15322

Sensory and motor coding in the organization of behavior

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZÜRICH

for the degree of
DOCTOR OF NATURAL SCIENCES

presented by
RETO WYSS

Dipl. Phys. ETH
born 22.3.1975
citizen of Meggen, Luzern

accepted on the recommendation of

Prof. Dr. Rodney Douglas
Dr. Paul Verschure
Prof. Dr. Peter König
Prof. Dr. Walter Senn

2003

Acknowledgment

First I would like to thank my parents Otto and Elisabeth Wyss for their great support, both financial and motivational, throughout the years of my studies. Special thanks belong to my bride-to-be Sascha - this thesis would not have been possible without the endless patience and encouragement I received from her.

This work has been carried out at the Institute of Neuroinformatics in the research group lead by Dr. Paul Verschure. Furthermore, I had the great pleasure of working closely together with Prof. Dr. Peter König. Both, Paul and Peter, kindly introduced me to the various aspects of scientific work and I owe to a large extent my current understanding of computational neuroscience to the many hours of discussions we had during the past few years.

Special thanks go to the students Roland Schätte, Tobias Kaufmann, Erik Vido, Philipp Knüsel, Peter Amberg, Armin Duff and Kay Jann whose work contributed either directly or indirectly to my thesis.

I would also like to acknowledge Fabian Roth, Andy Bähler, Philipp Knüsel, Gudrun Möller, Pamela Baker, and Aaron Schawalder with which I had the great pleasure of sharing the office with.

Further thanks go to the whole Institute of Neuroinformatics and in particular to Prof. Rodney Douglas and Prof. Kevan Martin who created a very pleasant and inspiring working environment.

Contents

Zusammenfassung	vii
Abstract	ix
1 Introduction	1
1.1 Temporal and spatial aspects of coding	1
1.2 Organization of this Thesis	6
2 The temporal population code	11
2.1 Introduction	11
2.2 Methods	13
2.2.1 Network	13
2.2.2 Clustering algorithm and mutual information	15
2.3 Results	16
2.3.1 Classification performance	16
2.3.2 Controls	21
2.4 Discussion	27
3 The enhanced temporal population code	31
3.1 Introduction	31
3.2 Methods	32
3.2.1 Network	32
3.2.2 Statistics	34
3.2.3 Stimuli	36
3.3 Results	37
3.3.1 Example of stimulus classification	37
3.3.2 Robustness of encoding	38
3.3.3 Code dimensionality	45
3.3.4 Scaling with the number of stimulus classes	48

3.4	Discussion	50
4	Motor system and decision making	53
4.1	Introduction	53
4.2	Methods	55
4.2.1	Sensorimotor mapping	56
4.2.2	Learning	57
4.2.3	Decision making	58
4.3	Results	59
4.3.1	Line following	59
4.3.2	Motor-map entropy	60
4.3.3	Crossing detection	60
4.3.4	Decision making	62
4.4	Discussion	67
5	Formation of place fields	69
5.1	Introduction	69
5.2	Methods	71
5.2.1	The temporal population code	71
5.2.2	Place cells from multiple snapshots	73
5.2.3	Position reconstruction	75
5.2.4	Place field shape and size	75
5.3	Results	76
5.3.1	Bounded invariance	76
5.3.2	Place cells from multiple snapshots	78
5.3.3	Position reconstruction	79
5.3.4	Place field shape	81
5.4	Discussion	83
6	Conclusion	87
A	The wSim simulation environment	93
A.1	Motivation - yet another neural simulator?	93
A.2	Design strategies	94
A.2.1	Polymorphism and templates	94
A.2.2	Data alignment and locality of information	95
A.3	wSim components	96

A.3.1	Library	96
A.3.2	Engine	98
A.3.3	Plugins	98
A.4	Example process	100

List of Figures

2.1	Schematic of the encoding paradigm	13
2.2	Network connectivity	14
2.3	Fisher Z-Transform	16
2.4	The simple stimulus set	17
2.5	Spike raster	17
2.6	Sample responses	18
2.7	Hit-matrices	19
2.8	Speed of encoding	21
2.9	Classification of handwritten characters	22
2.10	Synaptic noise	23
2.11	Transmission speed	24
2.12	Encoding of multiple stimuli	26
3.1	Schema of the network system	35
3.2	Three examples of the synthetic stimulus set	37
3.3	Sample responses	39
3.4	Robustness of encoding	41
3.5	The MNIST database	43
3.6	Clustering methods	44
3.7	Dendrograms and distance distributions	46
3.8	Principal component analysis	48
3.9	Information as a function of the number of stimulus classes	49
4.1	The micro-robot Khepera and the network model	55
4.2	Robot trajectories within the maze	60
4.3	Entropy in the motor-map	61
4.4	Sample crossing	63
4.5	Speed histograms	65
4.6	Maze coverage and inhomogeneity	66

5.1	Place cells from multiple snapshots	72
5.2	Tolerance with respect to viewing angle and distance . . .	76
5.3	Similarity surface for the four different cues	77
5.4	Similarity surface of Z cue for different reference points . .	78
5.5	Place fields of 5×5 place cells	80
5.6	Position reconstruction error	82
5.7	Place field asymmetry and size	83
A.1	wSim GUI-monitor screenshot	99

Zusammenfassung

Eine zentrale Frage der Neurowissenschaften ist, wie Informationen im Nervensystem kodiert sind. Während Neuronen untereinander mit diskreten singulären Signalen – sogenannten Spikes – kommunizieren, ist es unklar, ob die Information in der Feuerrate oder der präzisen zeitlichen Struktur der Spikes kodiert ist. Ähnlich unklar ist auch, in welchem Ausmass Populationen von Neuronen kooperieren oder Information unabhängig kodieren. Diese These versucht eine neue Sicht auf diese Aspekte zu erarbeiten indem eine neue Art von Kodierung vorgeschlagen wird welche beides beinhaltet, Kodierung im zeitlichen Bereich als auch kooperatives kodieren über eine Population von Neuronen. Während die Beurteilung einer Kodierungsstrategie eng von deren Einsatzgebiet abhängt, so legt die uniforme Anatomie des Kortex nahe, dass die Kodierung von Information in den verschiedenen Modalitäten ähnlichen Prinzipien folgen muss. Deshalb wird in dieser Dissertation grossen Wert darauf gelegt, die vorgeschlagenen Kodierungsverfahren in der echten Welt zu testen und vorallem verschiedene Modalitäten wie primäre sensorische, motorische und höher geordnete Repräsentationen in Betracht zu ziehen.

Im ersten Teil dieser Dissertation wird ein neuer Ansatz zur invarianten Mustererkennung präsentiert, der zeitliche Populationskode. Der Hauptbestandteil dieses Modells ist, dass Informationen im zeitlichen Bereich kodiert werden. Visuelle Stimuli werden durch ein kortikales Netzwerk von lateral gekoppelten Neuronen kodiert, wobei die systematische Abhängigkeit zwischen Verzögerung und Distanz der Kopplung von zentraler Bedeutung ist. Die resultierende Repräsentation des Stimulus ist positions- als auch rotationsinvariant wobei klassenspezifische Informationen bewahrt werden so dass eine Klassifizierung möglich bleibt. Der Kode ist robust im Bezug auf synaptisches Rauschen und kann sehr schnell erstellt werden. Zusätzlich können grosse Stimulussätze mit zunehmender Komplexität verarbeitet werden, wenn das Netzwerk mit orientierungs- sowohl als auch raumfrequenzselektiven Neuronen ausgestattet wird. So kann der zeitliche Populationskode erfolgreich zur Klassifizierung der standard MNIST Datenbank verwendet werden welche handgeschriebene Ziffern von 250 verschiedenen Schreibern enthält. Durch die Verwendung von grossen synthetisch generierten Stimulussätzen wird zudem gezeigt, dass das Konzept des zeitlichen Populationskodes auch für hunderte von Stimulusklassen anwendbar ist.

Um zu zeigen, dass sich dieses Kodierungsverfahren auch zu anderen Modalitäten verallgemeinern lässt, wird im zweiten Teil dieser Dissertation ein Model für die sensor-motorische Integration vorgestellt. Während die Verhaltenskontrolle aus traditioneller Sicht als eine Sequenz von sensorischer Verarbeitung, Entscheidungstreffung und Bewegungskontrolle beschrieben wird, so deuten neue experimentelle Studien darauf hin, dass das Motorsystem sehr wahrscheinlich für mehr als nur die Ausführung von Bewegungen verantwortlich ist. Auf diese Erkenntnisse aufbauend wird gezeigt dass die Populationsantwort des Motorsystems ein Substrat für die Klassifizierung von Verhaltenssituationen zur Verfügung stellt. Dieses Model verleiht der aufkommenden Rekonzeptualisierung der Verhaltenskontrolle weitere Glaubwürdigkeit. In diesem Zusammenhang kann das Motorsystem als ein höheres Wahrnehmungssystem aufgefasst werden, welches erlaubt wichtige Verhaltenssituationen zu erkennen, unabhängig von deren jeweiligen sensorischen Manifestation.

Nach erfolgreicher Anwendung in sensorischen als auch motorischen Systemen, stellt sich die Frage welche Eigenschaften des vorgeschlagenen Kodierungsverfahrens interessant für Representationen höherer Ordnung sein könnten. Im letzten Teil dieser Dissertation wird ein neues Model zur Erstellung von Ortszellen, welche im Hippocampus von Ratten gefunden wurden, vorgeschlagen. Dieses Model nützt die topologieerhaltenden Eigenschaften des zeitlichen Populationskodes aus. Aufgrund der relativ breiten Antwortcharakteristik dieser Ortszellen genügt eine kleine Anzahl um die Position mit zureichender Präzision zu bestimmen. Für einen sich verhaltenden Organismus bedeutet dies, dass eine flüchtige Erforschung einer neuen Umgebung bereits ausreicht um eine zureichende interne Representation zu erstellen.

Zusammengefasst wird gezeigt, dass das neue Kodierungsverfahren für verschiedene Modalitäten anwendbar ist wobei es interessante Invarianzeigenschaften aufweist welche nützlich für die Erstellung von höher geordneten Representationen der Welt sind.

Abstract

A central question in Neuroscience is how information is encoded in the nervous system. While neurons communicate with each other using spikes, i.e. discrete singular events in time, it is unclear whether information is effectively encoded in the average firing rates or the precise timing of single spikes. Similarly, it is not known in what context a population of neurons uses cooperative or independent coding. This thesis tries to elaborate a new view on these issues by proposing a new type of coding which features both, coding in the temporal domain as well as cooperative coding across neurons, the temporal population code. While the assessment of any coding strategy is strongly dependent on its range of applications, the uniformity of cortical anatomy suggests affinities between the coding of information of different modalities in cortical structures. Therefore, this thesis will be focused on applying the proposed coding scheme to real-world situations and in particular incorporating different modalities such as vision, motor and higher order representations.

In the first part of this thesis, a new approach towards invariant pattern recognition will be presented, the temporal population code. The key ingredient to the model presented is the coding of information in the temporal domain. Visual stimuli become encoded by a cortical network of laterally coupled integrate-and-fire neurons due to a direct relation between the transduction speed and the distance a signal has to travel between neurons. This representation of a stimulus is position- and rotation invariant while retaining class specificity in order to allow for their classification. The code is robust with respect to synaptic noise and can be generated quickly. Additionally, incorporating orientation and spatial frequency selective neurons allows to represent and reliably classify large stimulus sets of increased complexity. In this context, the temporal population code is successfully applied to the classification of the MNIST database, a benchmark database for pattern recognition systems containing handwritten digits from 250 different writers. Using large synthetically generated stimulus sets, it is shown that the concept of the temporal population code favorably scales to hundreds of stimulus classes.

In order to show that this coding strategy generalizes to other modalities, a model of sensori-motor integration is introduced in the second part of this thesis. While the traditional view on the control of behavior con-

siders a sequence of sensory processing, decision making and movement control, recent experimental studies suggest that the motor system is most likely more than only being responsible for motor execution. Building on these ideas it is shown, that the population response of the motor system provides a substrate for the categorization of behavioral situations. The model lends credence to the emerging reconceptualization of behavioral control. In this context, the motor system can be considered as part of a high-level perceptual system which allows for the detection of behaviorally salient situations invariant to their detailed sensory manifestation.

After successful application to the direct encoding of information in sensory as well as motor systems, the question arises, what properties of the proposed coding scheme could be of particular interest for higher order representations. In the last part of this thesis, a new model for the formation of place cells as found in rat hippocampus will be presented, which exploits the topology preserving properties of the temporal population code. Due to the relatively broad tuning of these place cells, only a relatively small number is required for accurate position reconstruction. Thus, for a behaving organism exploring an unknown environment, this implies that a relatively sparse exploration strategy suffices to create a complete representation of the new environment.

In summary, the new coding scheme considered has proven to be applicable to different modalities while providing interesting invariance properties for the formation of higher level representations of the world.

Chapter 1

Introduction

Over a century ago, neuroscientists have identified the single nerve cell as the elementary unit of processing in the nervous system [31, 65]. The postulate, however, that the brain as a whole relies in its function on extensively interconnected single units, raises the question, how these units communicate. A few decades later it was found that spikes, electrical impulses emitted by excited nerve cells, also called action potentials, constitute the substrate for most of the communication between nerve cells in the brain [3]. The remarkable feature of spikes is that they are highly stereotyped, varying only little from one nerve cell to another. Indeed, it was found that action potentials carried into the nervous system by a sensory axon are often indistinguishable from those carried out of the nervous system to the muscles by motor axons. Thus, the only features of the conducting signal of a single neuron conveying information is the frequency at which spikes occur and the interval between them. Considering a whole population of neurons, the locations at which spikes occur may also play an important role. From then until today, one of the key questions in neuroscience remains, how information may be encoded in the spiking activity of one single or a whole population of nerve cells.

1.1 Temporal and spatial aspects of coding

The first proposal for the type of coding employed by neurons goes back to the beginning of this century, where researchers became able to record from single cell fibers. Recordings in various sensory systems revealed that

the stimulus strength was directly related to the number of spikes a sensory neuron emits within a fixed time window [2]. Thus, this result indicated, that neurons code information in their firing rate and therefore apply a coding strategy which was called *rate coding*.

With the influential work of Hubel and Wiesel, the concept of rate coding has been tightly coupled to feature selectivity, the property of cells in sensory areas being selective for a small subspace of the complete sensory space [35]. They found that single cells in primary visual cortex (V1) respond selectively to bars of light presented within their receptive field, modulating their firing rate in relation to the orientation of the presented bar. Thus, as opposed to the early findings discussed above, which coined the term rate coding, cells in V1 do not only reflect the stimulation intensity in their firing rate but also other stimulus dimensions such as orientation. Here, already, the question arises why a cell would encode two parameters such as intensity and orientation in a single scalar value, i.e. its firing rate. Ambiguities introduced by such a strategy could be circumvented using time as an additional coding dimension. Indeed, experimental evidence is available which shows that cortical neurons can produce feature-specific phase lags in their activity [44].

The feature selectivity of V1 cells inspired a theory of visual processing which postulates that the visual system consists of a hierarchy of feature detectors, each of which becomes more and more sophisticated while moving up the hierarchy. One of the most influential views was given by Horace Barlow, according to whom visual scenes are represented by the activity of thousands of neurons, each being sharply tuned to a particular feature [12]. The system was said to employ a *local coding* strategy. Evidence for this type of coding has been found in higher visual areas, where cells have been found to respond selectively to for instance faces of particular individuals.

The assumption that cells are narrowly tuned to one particular feature accounts for the fact that cells that use rate coding may only code for a single scalar value. The very same assumption, however, gives rise to one of the most significant objections against the local coding theory; given the high dimensionality of a human's visual sensory space, the total number of distinct visual features would require by far more cells than what has been found in the visual system. Thus local coding can become very expensive, both in terms of the number of cells required and consequently also the necessary wiring between those cells. Therefore, although hierarchical

models of information processing have had a large success, new theories started to emerge, which aimed at attacking these problems, questioning either the rate coding or local coding hypothesis.

A shortcoming of local coding is the fact, that it does not truly exploit the representational capabilities of a neuron. Local coding requires, that a single cell simply signals the presence or absence of a particular feature, i.e. codes one bit of information. However, even under the assumption that the neurons in question use rate coding, more information could be provided taking graded responses into account. In particular, an assembly of cells, each coding for a different feature dimension, can cooperatively code for a complete sensory subspace. This coding strategy has been called *population coding* and has received most attention in the context of the motor system. Experimental studies have shown that parameters such as direction of movement are encoded by multiple cells in a distributed manner [29]. Thus, as an example, in order to code two-dimensional movements, local coding would require a whole matrix of cells each representing one single type of movement. The granularity, thereby, would be given by the number of cells within this matrix. Using population coding, however, only two cells would be required, each coding with graded responses for the movement along one of the two dimensions respectively.

With the advent of artificial neural networks (ANN), it became apparent, that distributed coding is not only a viable alternative to local coding. Distributed coding forms the foundation of distributed processing which was found to be computationally very powerful [73]. A very influential example was given by Hopfield and his auto-associative networks, also know as Hopfield networks [33]. These networks stressed two important and closely related concepts – recurrence and network dynamics. As opposed to the classical view on hierarchical networks by Barlow, which essentially operate in a feed-forward fashion, recurrent networks may display non-trivial network dynamics. Thus, these networks could not only process information distributed over space, but also over time. At this time, however, the precise evolution of the network state, and therefore the role of time in neural network function, was of secondary importance. Consequently, coding information in the temporal domain was generally not considered.

The first direct challenge of the rate coding hypothesis pointing towards the importance of temporal coding was given by a theoretical model pro-

posed by Milner [53]. It postulates that the degree of synchrony in the activity across a population of neurons provides an additional coding dimension, which could be used for tasks such as the segmentation of sensory scenes, also called *binding by synchrony*. In other words, sensory segmentation could be achieved dynamically by labeling the cells belonging to a coherent object by their synchronous firing. It is important to notice, that this model is not in conflict with hierarchical models of the visual system including the local coding hypothesis. This is due to the way information is represented in this type of encoding. For all the other types of encoding discussed so far, the encoded information is tightly coupled to its physical substrate, i.e. the neuron. In the case of binding by synchrony, however, the actual information is not tied to one particular physical unit, but rather lies in the coordinated activity of a population of neurons.

Binding by synchrony introduces an interesting new view on coding with respect to the neurons involved and their role in providing the physical substrate for encoding and representing information. In particular, the precise topological relationship between neurons partly loses its importance. Since the early work of Hubel and Wiesel, the notion of a receptive field has hardly changed and essentially describes the sensory area which elicits responses within a neuron upon stimulation. Closely related, the receptive field profile describes the spatial structure of the receptive field, i.e. it specifies, for example, the regions which excite or inhibit the cell. Traditionally, spatially structured receptive fields receive most attention in explaining response properties of primary sensory neurons. This is probably due to the precise topographical mappings found in primary sensory cortices. A neuron with a specific spatially structured receptive field profile can respond very selectively to a particular combination of sensory features and their topological arrangement within its receptive field. Thus, such neurons would employ a local coding strategy. The high selectivity, however, implies that they lack important invariance properties with respect to the stimulus position or other geometric parameters. These limitations are circumvented by coding strategies like binding by synchrony. The synchrony amongst a population of neurons is a code which does not depend on which neurons are involved and more importantly is invariant to their topological arrangement. In summary, a neuron reading out information from its receptive field may use topological or non-topological statistics, which essentially corresponds to spatially structured or uniform receptive

fields respectively.

A coding strategy which relies on non-topological statistics neglects all spatial information. This severe neglect of information leads to many invariances but also reduces the remaining coding space considerably. In the case of binding by synchrony, for example, only one single bit of information can be encoded, namely whether a group of neurons represents the same or different objects. To ameliorate this situation, a coding strategy based on non-topological statistics can make explicit use of the temporal domain. Instead of simply reflecting the temporal structure of the information to be encoded, a coding system may actively transform information from the spatial into the temporal domain. This has the advantage that the spatial information becomes detached from its precise physical substrate, the single neurons, and encoded in the temporal evolution of the coordinated activity of a whole population of neurons. Thus, given the mechanism responsible for transforming spatial information to the temporal domain, the resulting code becomes invariant to those spatial aspects which have not been transformed.

Building on these theoretical considerations, a new type of coding is introduced in this thesis, the *temporal population code*. It features both, coding in the temporal domain as well as cooperative coding across a population of neurons. A central ingredient to this proposal is the mechanism responsible for the generation of the temporal population code, and in particular the role it assigns to the massively recurrent and well ordered connections found in cortical structures [22]. The temporal population code arises from an active transformation of spatial information into a spatio-temporal representation. This transformation is performed by a recurrently coupled neural network. A subsequent transformation using non-topological statistics yields a purely temporal code, whose spatial invariances are determined by the symmetry properties of the transforming network. Thus, the network comprises both, the coding substrate as well as the coding mechanism which encapsulates local rules of computation from subsequent stages of processing.

In the context of a behaving system interacting with unknown and uncertain environments, any coding strategy has to conform to strict requirements allowing for fast and robust processing of relevant information. The temporal population code demonstrates, that non-topological statistics in combination with coordinated temporal coding across large populations of

neurons yields a powerful coding strategy, which exhibits many advantages over other approaches. For instance, the speed of information processing by systems using rate coding is limited since the frequency of a neuron can only be estimated taking several spikes into account. With the temporal population code, however, a large part of the encoded information can already be conveyed by as little as two spikes from each individual neuron. Another approach towards the efficient usage of limited processing resources consists of processing only information relevant to the system while discarding the rest, i.e. for pattern recognition, the stimulus position is not important and can therefore be discarded. This strategy is adopted by the temporal population code using non-topological statistics, i.e. to neglect all information not transformed into the temporal domain by the encoding mechanism.

Finally, in order to be able to interact with a real-world environment in a continuous manner, internal representations should reflect the sensory topology, i.e. similar sensory information is mapped to similar representations. From another point of view, given that a behaving agent is moving consistently in space and time, the global sensory percept will change on a behavioral time scale. Therefore, the internal representations of the world should also vary on the behavioral rather than the neural time scale. This view is also supported by recent theoretical studies suggesting that the “slowness” of neural responses might be a general objective of neural information processing systems [14,39,98]. As will be shown in this thesis, the temporal population code exhibits this topology preserving property which will be exploited for the formation of allocentric higher order representations of environments from purely local egocentric sensory information.

1.2 Organization of this Thesis

This thesis is organized in six chapters, whereas the central four chapters represent the main body. The second and third chapter combines the material of three publications [100, 101, 103]. Each of the two subsequent chapters is based on an article either published or accepted for publication [99,102]. Most of the results have also been presented at conferences or/and appeared in abstract form in conference proceedings. The different articles have not been incorporated into this thesis in their original format, but have

been partially rewritten and reorganized in order to prevent unnecessary repetitions.

In the second chapter, a new approach towards invariant pattern recognition will be presented, which is based on a temporal population code. A number of models have been proposed which aim to reproduce capabilities of the visual system such as invariance to shifts in position, rotation and scaling. Most of these models are based on the Neocognitron, a hierarchical multi-layer network of spatial feature detectors. These models do not consider time as a coding parameter, although recently, the importance of the temporal dynamics of neuronal activity in representing visual stimuli has gained increased attention. The key ingredient to the model presented in this thesis is the coordinated coding of information in the temporal domain across large populations of neurons. Visual stimuli become encoded by a cortical network of laterally coupled integrate-and-fire neurons due to a direct relation between the transduction speed and the distance a signal has to travel between neurons. This representation of a stimulus is position- and rotation invariant while retaining class specificity in order to allow for their classification. The code is robust with respect to synaptic noise and can be generated quickly.

While the above model incorporates a rather uniform network topology, a second, more elaborate model featuring cortical cells with orientation as well as spatial frequency selectivity is presented in the third chapter. The connectivity between cells is no longer isotropic but depends on their relative spatial frequency as well as orientation selectivities. This advanced network can cope with stimulus sets of increasing complexity, both in terms of variability between stimuli belonging to the same stimulus class as well as the number of stimulus classes which need to be distinguished simultaneously. In this context, the temporal population code is successfully applied to the classification of the MNIST database, a benchmark database for pattern recognition systems containing handwritten digits from 250 different writers. In addition, using large synthetically generated stimulus sets, it is shown that the concept of the temporal population code favorably scales to hundreds of stimulus classes. Finally, several different experiments are described which aim at determining the dimensionality of the enhanced temporal population code.

In contrast to the classical notion of a receptive field, the temporal population code neglects any spatial topology. It rather considers a non-

topological statistical measure on the activity of a whole population of neurons and the evolution of this measure over time. In order to show that this coding strategy generalizes to other modalities, a model of sensori-motor integration is introduced in the fourth chapter of this thesis. This model, implemented on an autonomous mobile robot, uses such a population code to identify behaviorally important situations. While the traditional view on the control of behavior considers a sequence of sensory processing, decision making and movement control, recent experimental studies suggest that the motor system is most likely more than only being responsible for motor execution. Building on these ideas it is shown, that the population response of the motor system provides a substrate for the categorization of behavioral situations. This categorization allows for the assessment of the complexity of a behavioral situation and regulates whether higher level decision making is required in order to resolve behavioral conflicts. The model lends credence to the emerging reconceptualization of behavioral control. In this context, the motor system can be considered as part of a high-level perceptual system which allows for the detection of behaviorally salient situations invariant to their detailed sensory manifestation.

After successful application to the direct encoding of information in sensory as well as motor systems, the question arises, what properties of the proposed coding scheme could be of particular interest for higher order representations. Thus, the objective of the fifth chapter is to investigate to what extent the properties of a temporal population code can be exploited for the formation of place cells, i.e. cells found in the hippocampus which respond selectively for specific positions within an environment. Unlike other approaches towards invariant pattern recognition, the temporal population code naturally generalizes across different views of a stimulus; their visual similarity is directly accessible through the similarity of their representations without the need for learning. In the context of an agent behaving in an environment with different visual cues, this means that the similarity amongst different viewing angles and distances directly translates into the relative locations within the environment where these views are perceived. This property allows to construct place cells from multiple temporal population code snapshots of the environment taken at the same location but different orientations. Due to the relatively broad tuning of these place cells, only a relatively small number is required for accurate position reconstruction. Thus, for a behaving organism exploring

an unknown environment, this implies that a relatively sparse exploration strategy suffices to create a complete representation of the new environment.

Finally, in the general conclusions the main results of this thesis are summarized and limitations to the presented models are discussed.

Chapter 2

The temporal population code

2.1 Introduction

Mammals demonstrate highly evolved visual object recognition skills, tolerating considerable changes in images due to, for instance, different viewing angles and deformations. Elucidating the mechanisms of such *invariant pattern recognition* is an active field of research in neuroscience [25, 37, 47, 71, 84]. However, still very little is known about the underlying algorithms and mechanisms. A number of models have been proposed which aim to reproduce capabilities of the biological visual system, such as invariance to shifts in position, rotation and scaling [62, 69, 96]. Most of these models are based on the *Neocognitron* [26], a hierarchical multi-layer network of spatial feature detectors. As a result of a gradual increase of receptive field sizes, translation invariant representations emerge in the form of activity patterns at the highest level. These models do not consider time as a parameter of neural representations. Recently, however, the importance of the temporal dynamics of neuronal activity in representing visual stimuli has gained increased attention [11, 27, 79]. Hence, it seems timely to consider the role of temporal coding in the context of tasks like invariant object recognition.

In recent years, several modeling studies have addressed properties of temporal codes [9, 34, 91]. For instance, Buonomano and Merzenich proposed a model for position-invariant pattern recognition which uses tem-

poral coding [17]. In this model, feed-forward inhibition modulates the spike-timing such that stimuli are represented by the response latencies of the neurons in the network. This architecture naturally leads to translation invariant representations. This model assigns a critical role to inhibitory interactions in the feed-forward path of the visual system (retina-LGN-V1), whereas anatomical studies suggest that these connections are predominantly excitatory [38]. Furthermore, the majority of inputs to cortical neurons are excitatory and of cortical origin [22]. Indeed, a recent theoretical study has shown that lateral excitatory coupling has pronounced effects on the global network dynamics [93]. In particular, the combination of intracortical connectivity and dendritic processing allowed context dependent representations of different stimuli, to be expressed in the temporal dynamics of the network. Here we build on these previous proposals and concepts [17] and investigate the formation of invariant representations by the dynamics of activity of neuronal populations.

In this chapter, we investigate a model of primary visual cortex consisting of a map of integrate-and-fire neurons with lateral excitatory interactions. A central feature of this model is the monotonic relationship between transmission delays in this lateral coupling and the distance between pre- and postsynaptic neurons. We hypothesize that this network property induces dynamics of neuronal activity (fig. 2.1a) that are specific to the geometry of a stimulus and invariant with respect to several transformations. The advantage of such a representation is that it emerges naturally without the need of training the network repeatedly for different stimulus positions or orientations. In order to investigate the validity of this hypothesis, we determine the amount of information contained in the temporal population responses of this network for different parameters and stimulus sets. Furthermore we investigate the speed of encoding and its robustness to synaptic noise. The results suggest that invariant pattern recognition can be achieved using temporal coding at the population level.

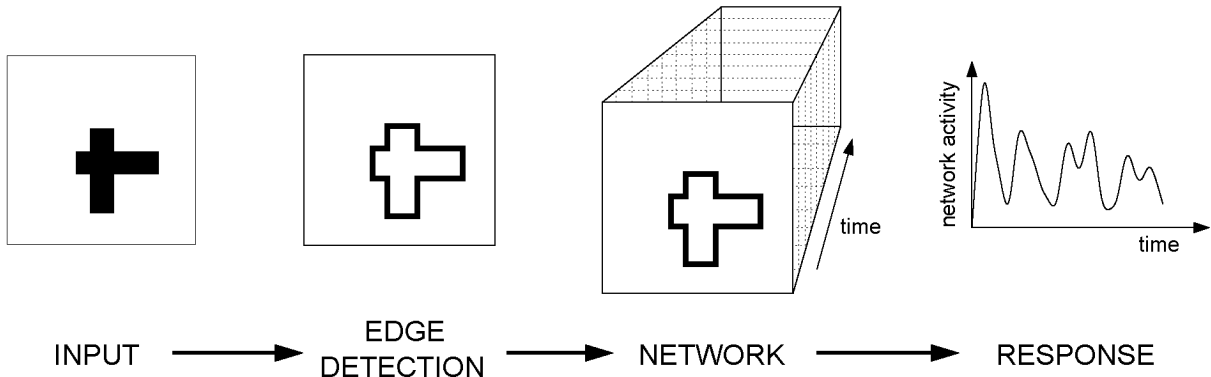


Figure 2.1: Schematic of the encoding paradigm. “Solid” input-patterns pass through an edge-detection stage and the resulting contour is projected topographically onto a map of cortical neurons. Due to the lateral intracortical interactions, the stimulus becomes encoded in the network’s activity-trace.

2.2 Methods

2.2.1 Network

The investigated network consists of a two-dimensional array of 40×40 conductance-based leaky integrate-and-fire neurons, which include a spike-triggered potassium conductance yielding frequency adaptation. Under constant excitation and after adaptation, these neurons spike regularly. The time course of a leaky integrate-and-fire neuron’s membrane voltage $V(t)$ is described by the differential equation:

$$C_m \frac{dV}{dt} = -(I_{exc}(t) + I_{inh}(t) + I_K(t) + I_{leak}(t)) \quad (2.1)$$

where C_m is the membrane capacitance ($C_m = 0.2$ nF), and I represents the transmembrane current, i.e. excitatory input (I_{exc}), inhibitory input (I_{inh}), spike-triggered potassium current (I_K) and leak current (I_{leak}). These currents are computed by multiplying a conductance g with the driving force: $I(t) = g(t)(V(t) - V^{rev})$ where V^{rev} is the reversal potential of the conductance ($V_{exc}^{rev} = 60$ mV, $V_{inh}^{rev} = -70$ mV, $V_K^{rev} = -90$ mV, $V_{leak}^{rev} = -70$ mV). The neuron’s activity at time t , $A(t)$, is given by $A(t) = H(V(t) - \theta)$ where H is the Heaviside function and θ is the firing threshold ($\theta = -55$ mV).

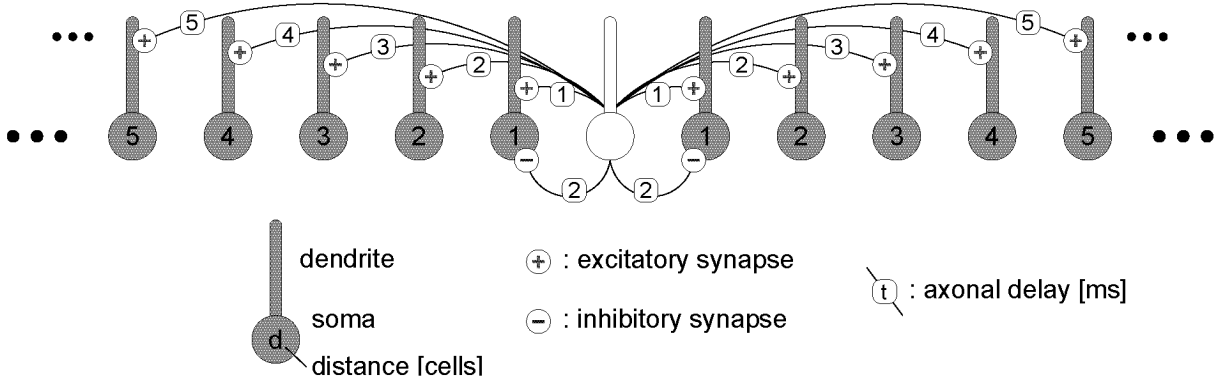


Figure 2.2: Two-dimensional view of the network connectivity. Each neuron connects through long-range excitatory connections to its neighbors whereas the transmission delay τ increases linearly with the distance the signal has to travel, i.e. $\tau = 1$ ms/cell. For one control experiment, in addition, the cells were connected with their nearest neighbors through inhibitory synapses and a delay of 2 ms.

Each time a spike is emitted, the neuron’s potential is reset to $V_{rest} = V_{leak}^{rev}$. The constant leak conductance g_{leak} is 20 nS. The time course of the potassium conductance is given by $\tau_K dg_K/dt = -(g_K(t) - g_K^{peak} A(t))$ where $A(t) \in \{0, 1\}$ with a time constant τ_K and a peak conductance g_K^{peak} ($\tau_K = 40$ ms, $g_K^{peak} = 200$ nS). The synaptic interactions are “instantaneous”, such that the total synaptic conductance at time t is the linear sum over all active conductances derived from the individual synapses at time t . In the discrete-time simulations, the equations above are integrated with Euler’s method and a temporal resolution Δt of 1 ms.

Each neuron connects to a circular neighborhood of fixed size, such that neurons with Euclidean distance ≤ 9 cells are connected (fig. 2.2). The synapses are of equal strength ν and are modeled as instantaneous excitatory conductances, while transmission delays are related to the Euclidean distance between the positions of the pre- and postsynaptic neurons with a proportionality factor of $\tau = 1$ ms/cell (fig. 2.2). Additionally, for one control experiment, each cell connected to its nearest neighbors through an inhibitory synapse with a transmission delay of 2 ms. Stimuli are presented continuously to the network and first pass through an edge-detection stage (LGN) and the resulting contours are projected topographically onto the

array of neurons (V1) using a tonic excitatory input conductance (fig. 2.1). After frequency adaptation, the stimulated neurons spike at a frequency of approximately 42 Hz.

2.2.2 Clustering algorithm and mutual information

The algorithm for clustering the responses of the network is adapted from Victor and Purpura [94]. The network's responses to stimuli from C stimulus classes S_1, S_2, \dots, S_C are assigned to C response-classes R_1, R_2, \dots, R_C yielding a $C \times C$ hit-matrix $N(S_\alpha, R_\beta)$, whose entries denote the number of times that a stimulus from class S_α elicits a response in class R_β . Initially, the matrix $N(S_\alpha, R_\beta)$ is set to zero. For each response, $r \in S_\alpha$, we calculate the average temporal correlation of r to the responses $r' \neq r$ elicited by stimuli of class S_γ :

$$\bar{\rho}(r, S_\gamma) = Z^{-1} \left(\left\langle Z(\rho(r, r')) \right\rangle_{r' \text{ elicited by } S_\gamma} \right) \quad (2.2)$$

where $\rho(r, r')$ is the temporal correlation between r and r' , $\langle \cdot \rangle$ denotes the average. Z is the Fisher Z-Transform given by $Z(\rho) = 1/2 \ln((1 + \rho)/(1 - \rho))$, which transforms a distribution of correlation coefficients ρ into an approximately normal distribution of coefficients, $Z(\rho)$ (fig. 2.3). Thus, $Z(\rho)$ becomes a measure on a proportional scale such that mean values are well defined. The average correlation is also computed for the stimulus class S_α which elicited r , but since $r \neq r'$, the term $\rho(r, r)$ is excluded from (2.2). The response r is classified into the response-class R_β for which $\bar{\rho}(r, S_\beta)$ is maximal, and $N(S_\alpha, R_\beta)$ is incremented by one. If k $\bar{\rho}$'s share the maximum, each corresponding matrix-element is increased by $1/k$.

An information-theoretic measure, the *mutual information* I , quantifies the extent to which this clustering is random. For stimuli that are drawn from discrete classes S_1, S_2, \dots , and responses that have been grouped into discrete classes R_1, R_2, \dots , the mutual information I is given by

$$I = \frac{1}{N_{tot}} \sum_{\alpha, \beta} N(S_\alpha, R_\beta) \left[\log_2 N(S_\alpha, R_\beta) + \log_2 N_{tot} - \log_2 \sum_a N(S_a, R_\beta) - \log_2 \sum_b N(S_\alpha, R_b) \right]$$

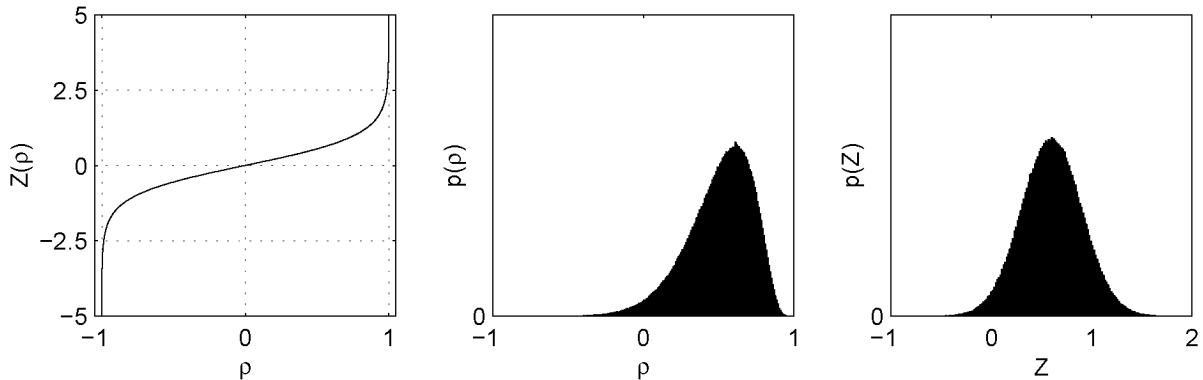


Figure 2.3: The Fisher Z -Transform (left) transforms a skewed distribution of correlation coefficients ρ (middle) into an approximately normal distribution $Z(\rho)$ (right).

where N_{tot} is the total number of stimuli. For C equally probable stimulus classes, random classification corresponds to $N(S_\alpha, R_\beta) = N_{tot}/C^2$ for $\forall \alpha, \beta \in \{1, \dots, C\}$, where I becomes zero. For perfect classification, where each diagonal element of $N(S_\alpha, R_\beta)$ is equal to N_{tot}/C , the mutual information becomes maximal, i.e. $I = \log_2 C$.

2.3 Results

2.3.1 Classification performance

In a first experiment, we investigate the concept and test the basic network's performance in the invariant encoding of hand-drawn stimuli. The arrangement of the six stimulus classes (fig. 2.4a) reflects an intuitive notion of topology, i.e. class 1 is visually more similar to class 2 or 3 than to class 5 or 6. For each stimulus class, 24 samples are presented to the network (fig. 2.4b).

The spike raster plot for a sample of class 1 illustrates the network activity (fig. 2.5a). An initial synchronous phase is followed by a dispersion of activity. Such raster plots give a good description of network activity. However, because activities of such large numbers of neurons are rarely recorded simultaneously in visual cortex, they do not allow a direct comparison to physiological data. Therefore, population averaged histograms

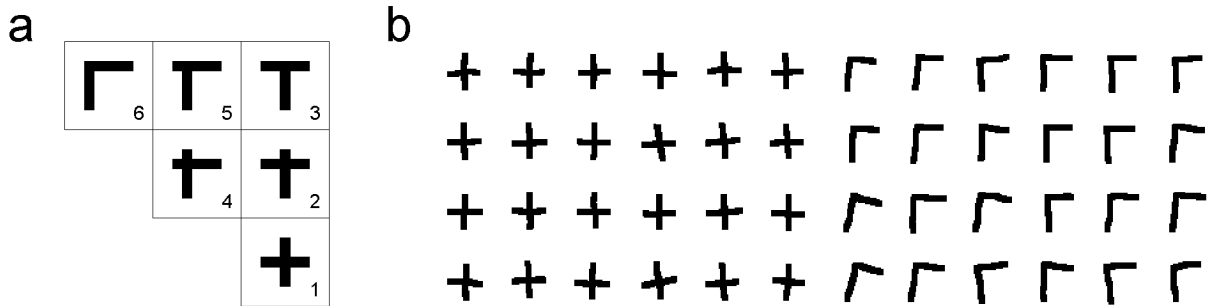


Figure 2.4: The stimulus classes. **(a)** Each stimulus class consists of a horizontal and a vertical bar of equal length, but intersecting at different locations for the individual classes. The arrangement reflects an intuitive notion of topology in a sense that visually similar stimulus classes are closer than dissimilar ones. **(b)** From each stimulus class, 24 hand-drawn samples were presented to the network.

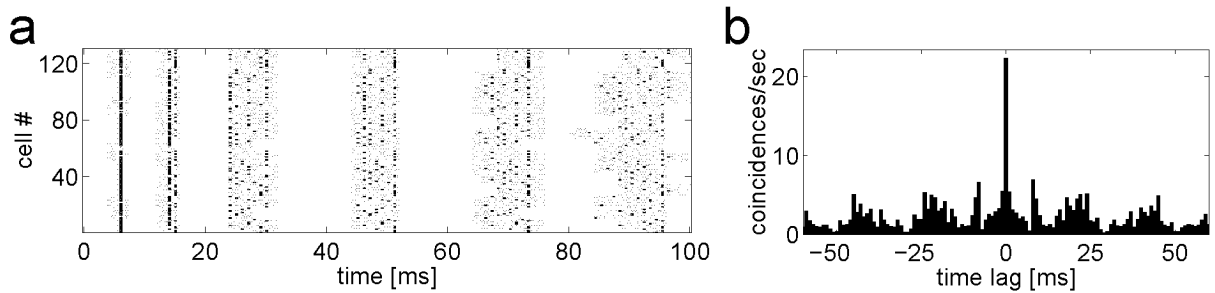


Figure 2.5: **(a)** Spike raster of 130 neurons of the network while presenting a stimulus sample of class 1. **(b)** The population cross-correlation function for the neurons shown in panel (a).

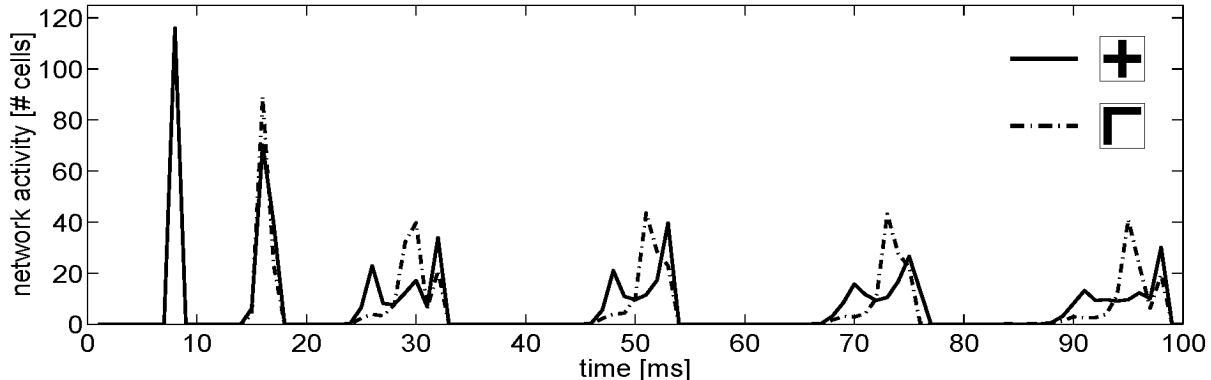


Figure 2.6: Sample responses of the network to stimulus class 1 and 6.

are not used to describe physiological data but rather time averaged histograms [30]. For comparison we compute the multi-unit cross-correlation function (fig. 2.5b). The central peak signifies the population bursts and the satellite peaks are a sign of the temporally structured activity. Overall, this cross-correlation function resembles those acquired in the visual cortex of mammals [43, 78].

Qualitatively, the difference between the responses of the network to different stimulus classes is shown in fig. 2.6. After the initial identical synchronous phase, the dispersion of the population activity over time is different for the two stimulus classes.

For a quantitative analysis, the responses of the network are clustered into six classes using the temporal correlation of the population response as a similarity-measure. For each stimulus the correlation is computed with all other responses over an interval of 100 ms. It is assigned to the response class which maximizes the average of this measure (see sec. 2.2.2). The resulting hit-matrix (fig. 2.7a) reveals that 91% of the stimuli are classified correctly (diagonal entries). The misclassifications (9%, non-zero off-diagonal entries), however, do show some regularity. They only occur among visually similar stimuli. Indeed, more than half of the misclassifications result from a confusion of the classes 4 and 5 which only differ in the precise position of the vertical bar. In order to investigate the relationship between the strength of the lateral coupling, ν , and the encoding performed by the network, we vary this parameter in the range of $0 < \nu < 0.25$ nS. For each of these conditions we calculate the *mutual information* from the

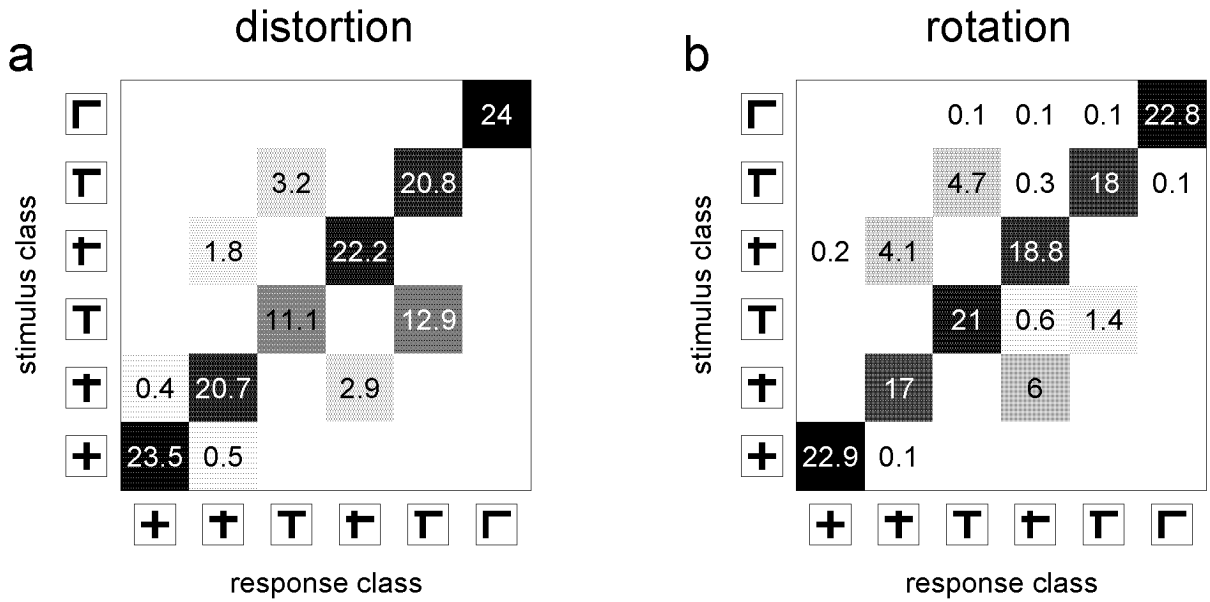


Figure 2.7: Hit-matrices. The matrix entries represent the number of times a stimulus class (fig. 2.4a) is assigned to a response class, averaged over 24 trials. **(a)** Distortion stimulus set with 24 samples per stimulus class and $\nu = 0.13$ nS ($\hat{=} 2.1$ bits). **(b)** Rotation stimulus set with 23 samples per stimulus class and $\nu = 0.1$ nS ($\hat{=} 2.1$ bits).

hit-matrix (see Methods). We find that for $\nu < 0.01$ nS the activity-traces contain no information about the input stimuli. For $0.01 < \nu < 0.25$ nS, however, mutual information reaches 1.77 ± 0.21 bits (mean \pm std, of $\log_2(6) = 2.6$ bits possible) and all hit-matrices qualitatively resemble the one shown in fig. 2.7a. In summary, the network successfully and reliably encodes the six stimulus classes in a large range of lateral coupling strength.

In a next step we investigate the classification of rotated stimuli. A stimulus set is constructed by taking one sample out of each stimulus class (fig. 2.4a), and generating the complete sets by rotating each sample by 23 evenly spaced angles between 0° and 360° ¹. As in the previous experiment, no information about the input is conveyed by the network’s activity traces for $\nu < 0.01$ nS. For $0.01 < \nu < 0.25$ nS, mutual information reaches 1.56 ± 0.29 bits. Maximal information of 2.1 ± 0.04 bits is attained for $\nu = 0.1$ nS. The corresponding hit-matrix (fig. 2.7b) shows the same tendency as for the distortion stimulus set: 87% of the stimuli are classified correctly, while 12% are confused with nearest neighbors and only 1% with non-nearest neighbors. Hence, the population activity-trace reliably encodes the six stimulus classes invariant to rotation.

In order to analyze the speed of encoding we determine the amount of information encoded in the network’s activity-trace at different times after stimulus onset. Varying the length of the interval used to compute the correlation between different responses between 2 to 100 ms we observe that 66% of the information is available after 20 ms (fig. 2.8). This property of our encoding scheme is compatible with the impressive speed of processing found in the mammalian visual system [86].

How the present model scales with an increasing number of stimulus classes was the subject of the following experiment. A handwritten (single writer) uppercase Roman alphabet was presented to the network, with 26 stimulus classes with 20 samples each (fig. 2.9a). For $0.02 < \nu < 0.25$ nS, information reaches 3.92 ± 0.12 bits (of $\log_2(26) = 4.7$ bits possible). Maximal information of 4.16 ± 0.02 bits is attained for $\nu = 0.1$ nS, where 88% of the stimuli are classified correctly. The corresponding hit-matrix is shown in fig. 2.9b. Thus the proposed encoding scheme scales to problems

¹Since the whole system is manifestly invariant under rotations by multiples of 90° , these rotations have to be omitted, otherwise the task would become trivial. Therefore, the number of angles should not be a multiple of 4.

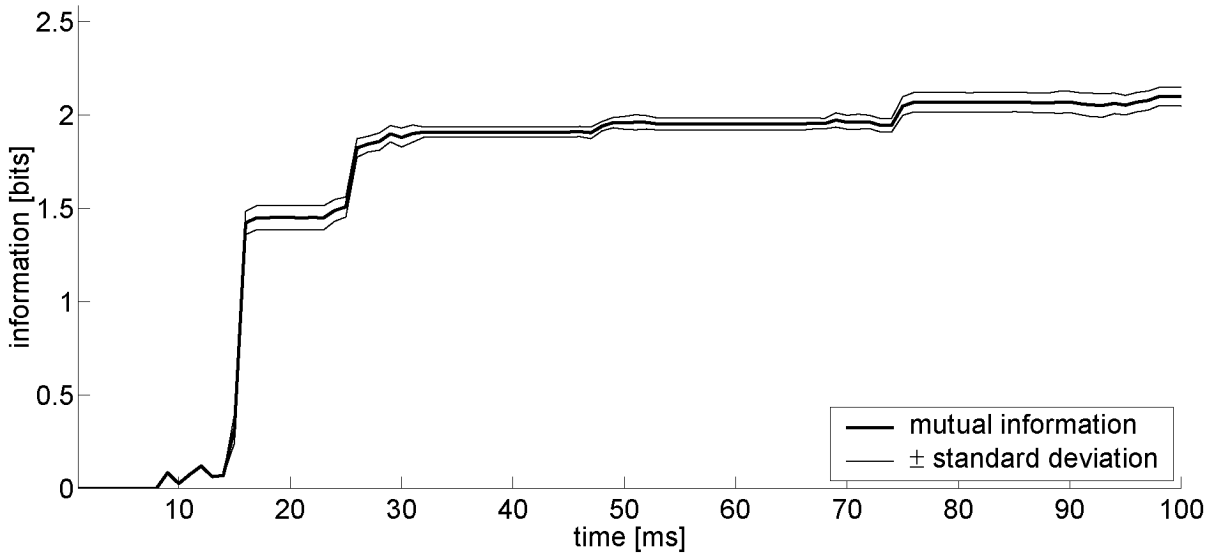


Figure 2.8: Average information encoded in the network’s activity traces as a function of time for the distortion stimulus set with $\nu = 0.13$ nS. The standard deviation is calculated over 24 trials.

of interesting complexity.

2.3.2 Controls

It has been argued that a potential problem of encoding information in the timing of action potentials is the highly unreliable transmission of signals across synapses [6]. Thus, we investigate the robustness of the proposed encoding scheme with respect to synaptic noise in the lateral coupling. Synaptic noise is modeled by perturbing the individual synaptic conductances dynamically; each conductance is multiplied by a random factor f , drawn from a normal distribution with mean one and variance σ^2 , i.e. $f \in N(1, \sigma^2)$. In order to prevent negative conductances, the normal distribution is clipped at zero. The system’s performance in encoding the input stimuli decreases linearly with increasing noise (fig. 2.10a). The number of correctly classified stimuli decreases by not more than 25% for $\sigma = 1$, which corresponds to a signal to noise ratio of 1. Furthermore, misclassifications are to a substantial part (90% for $\sigma = 1$) due to the confusion of nearest neighbors (fig. 2.10b). Thus, the information encoded at the population

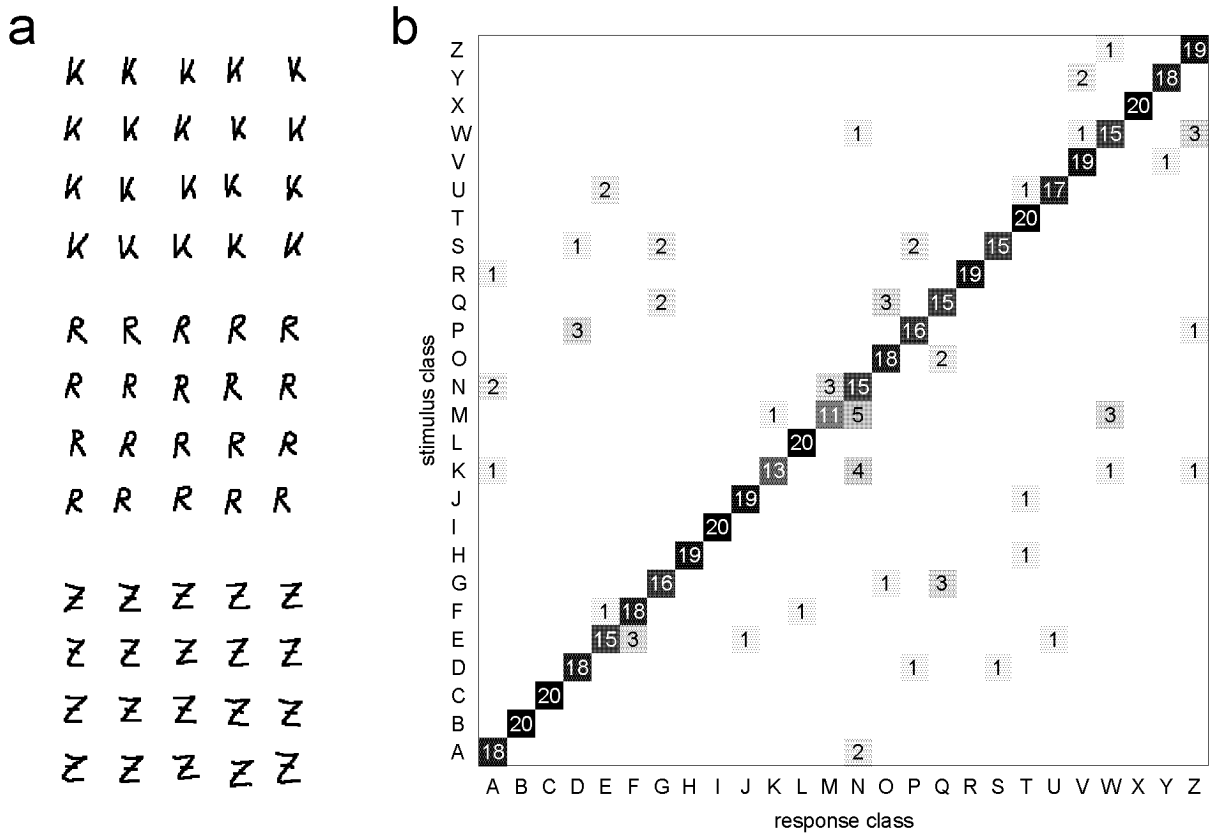


Figure 2.9: Classification of handwritten characters. **(a)** Stimuli of the classes K, R and Z as examples from a single writer presented to the network. Each stimulus class contains 20 samples. **(b)** The resulting hit-matrix for $\nu = 0.1$ nS (single trial).

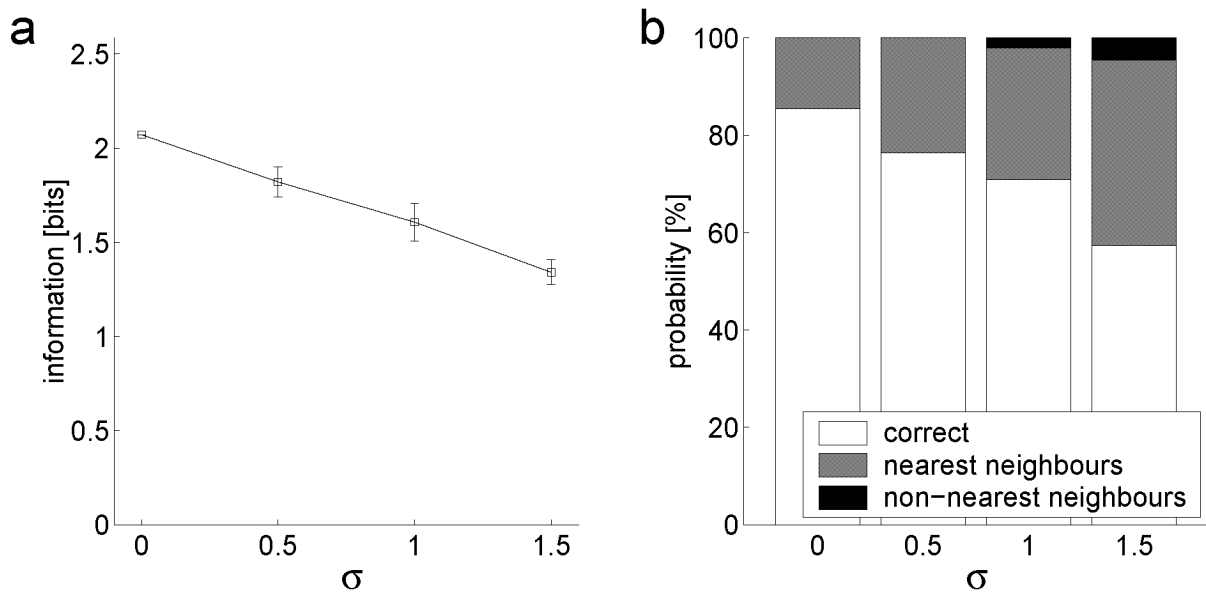


Figure 2.10: Influence of synaptic noise on encoding. **(a)** Information as a function of the noise level σ for the distortion stimulus set with $\nu = 0.13$ nS, averaged over 24 trials; error bars indicate standard deviation. **(b)** The probability of a stimulus being classified either correctly, to a nearest neighbor or to a non-nearest neighbor for different noise levels σ .

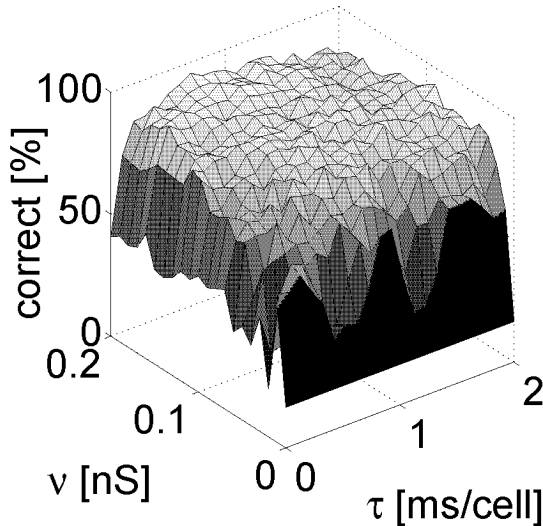


Figure 2.11: Correctly classified stimuli as a function of the inverse of transmission speed and synaptic strength of the lateral coupling.

level is robust in respect to synaptic noise.

In the following experiment we would like to investigate the network's encoding performance for different speeds of tangential interactions. For this purpose, the temporal resolution of the simulation was increased to $\Delta t = 0.1$ ms, allowing for the delay slopes τ to take values from 0 to 2 ms/cell in 0.1 ms/cell steps. For the clustering we binned the network responses with a temporal resolution of 1 ms. In addition, the synaptic strength ν was varied simultaneously in the range from $0 < \nu < 0.2$ nS in order to detect combined effects of τ and ν on encoding performance. As it is shown in fig. 2.11, the performance does not significantly depend on τ , as long as the latter is above ≈ 0.3 ms/cell. Furthermore, apart from weak synaptic strengths (i.e. $\nu < 0.04$ nS) we do not observe any systematic relationship between ν and τ . Thus the encoding scheme proposed here is remarkably invariant to the choice of two defining parameters of the lateral coupling, which constitutes the central component of the proposed network leading to the transformation of the spatial stimulus into the temporal domain.

As a further control we study the encoding of random stimuli. We present random dot patterns which have the same pixel density as the bar-stimuli investigated before. The random stimuli are compared to the

clusters of the bar stimuli from the first experiment. We find that the distance of the random stimuli from these clusters, in units of the width of the distribution of the bar-stimuli, is large (29.8 ± 2.4 , mean \pm std, $n=500.000$). Hence, with a reasonable choice of a classification threshold, e.g. a distance of at most 3 standard deviations resulting in approximately 0.5% false rejections, the number of false positives is virtually zero. Thus, it is unlikely that in the proposed encoding scheme random stimuli are confused with structured stimuli.

So far, the network we have investigated only incorporates excitatory coupling between neurons, while inhibitory interactions, which are ubiquitous throughout the cortex, have not been considered. Hence, as a control we add lateral inhibitory connections to our network model. Each cortical neuron inhibits its direct neighbors in the map with a synaptic conductance of 10 nS and a delay of 2 ms (fig. 2.2). We find that for all stimulus sets presented above, there is no significant difference in the system's performance, i.e. base set: 1.78 ± 0.19 bits, rotation set: 1.5 ± 0.27 bits. These results indicate that the proposed encoding scheme is not affected by incorporating inhibitory interactions in the neural network.

The effect of the presence of multiple stimuli is addressed as a further control. We investigate a four times wider network. The activity induced by a target stimulus is evaluated by pooling neuronal activity within a large region (readout region), which however does not encompass the whole network. Two randomly chosen stimuli serve as distractors on either side (fig. 2.12a). When the readout region of the stimulus includes the distractors a small decrease in performance is observed (fig. 2.12c). If by virtue of the tangential connections in the network the distractors interact with the representation of the target stimulus the encoded information is more reduced (fig. 2.12b). A combination of both effects leads to a more severe interference (fig. 2.12d). Note, however, that this situation creates a continuous stimulus pattern that poses severe problems for any recognition system. Furthermore, given the scaling properties of the network (see sec. 3.3.4) it is possible to encode pairs of neighboring stimuli as one compound stimulus. Hence, the presence of multiple stimuli does not pose a fundamental problem to the encoding scheme investigated here.

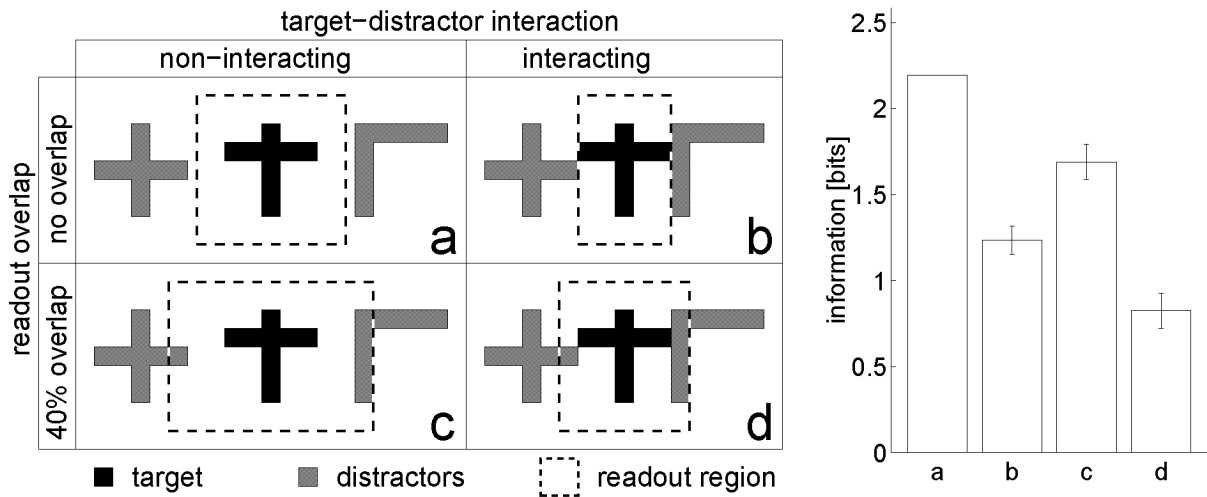


Figure 2.12: Multiple Stimuli. The panels a-d show the target stimulus (black), the distracting stimuli (gray) and the readout region (dashed rectangle) in four characteristic situations. **(a)** Control condition: stimuli do not interact and readout only captures target stimulus. **(b)** Representations of the stimuli interact through the lateral coupling within the network. **(c)** Readout captures activity from the distracting stimuli. **(d)** Conditions (b) & (c) are combined. The histogram on the right shows the performance of the system in encoding and classifying stimuli in the four situations resp. (errorbars: \pm standard deviation over 20 trials).

2.4 Discussion

We have shown that, in a model of a cortical network, the interaction of network- and stimulus-topology induces stimulus-specific but transformation invariant temporal dynamics. Thus, stimuli are represented by a temporal population code. This representation is position- and rotation-invariant while reasonably robust to stimulus variability. The stimulus encoding preserves an intuitive notion of visual similarity. Furthermore, it is robust with respect to synaptic noise. These properties apply in a range of the strength and speed of the lateral coupling each spanning more than one order of magnitude.

In the study presented in this chapter, we tried to understand how key functional properties of the visual system can be accounted for by a temporal population code. Compared to area 17, however, our model contains several assumptions. It is composed of conductance-based integrate-and-fire neurons, which are coupled through excitatory connections. The key assumption of our model is that the transduction latencies of these connections depend systematically on the distance between pre- and postsynaptic neurons. Indeed, indirect evidence for such a relationship has been found [54, 85]. These transduction delays could result from dendritic delays due to the distance a signal has to travel between the synapse and the soma of the postsynaptic neuron [4]. Thus, our study predicts that a positive correlation exists between the separation of receptive fields and the distance between the dendritic location of the respective synapses and the soma.

The speed of information flow found in biological visual systems poses severe constraints upon computational models of pattern recognition. Experimental studies have shown that the analysis and classification of complex visual patterns can be carried out by humans in not more than 200 ms [87]. Considering that a minimum of 10 areas are involved to reach the relevant processing areas from the retina, little time is left for intra-areal processing. This view is supported by studies in macaque monkeys [89, 90], which show that single visual areas can process significant amounts of information in just 20 to 30 ms. This is compatible with our finding that 66% of information about a stimulus becomes encoded within 20 ms. The gradual increase of encoded information over time we observed indicates that subsequent processing stages can be engaged before all the information

about a stimulus has been encoded. Indeed, experimental data on signal timing in the macaque visual system show that the distribution of onset latencies of different visual areas overlap substantially [76]. This suggests that the dominant mode of processing in the visual system is concurrent rather than strictly feed-forward.

A number of studies have proposed a role for temporal dynamics in the formation of visual representations. One of the first proposals [66] suggested that local features of visual stimuli are coded in the temporal patterns of spikes of single neurons. Based on principal component analysis of the response patterns of single neurons, a significant amount of information carried by the first (\cong firing rate) as well as the higher components (\cong temporal pattern) was found [90]. An important difference with the present study is that we consider temporal coding at the population level, rather than at the level of a single neuron. Recently, Buonomano and Merzenich presented a model on the generation of temporal population codes [17] which contained orientation selective feature detectors and strong feed-forward inhibition. It was shown that the latencies between stimulus onset and the first spike of the neurons in the network constitute a representation which is invariant to the position of presented stimuli. In contrast, by relying on lateral interactions, the present model accumulates information over time which results in reliable encoding.

As a particular example of a temporal population code, synchronized neuronal activity has been intensively studied [1, 11, 79]. A large number of experiments have reported synchronized activity in a variety of species and cortical structures [78, 80]. In particular, it has been argued that the synchronization of neural activity provides a substrate for the binding and segmentation of visual patterns [53, 95]. Furthermore, it has been shown that the synchronization of neuronal activity is mediated by intracortical connections without changing receptive-field properties of the postsynaptic neurons [23, 57]. Thus, synchronous activity does not contain information about stimulus features as such and can be seen as a binary signal about global stimulus properties. Our model shows that synchronous activity at a broad time scale combined with dispersion on a fine time scale provides a high-dimensional signal in a temporal population code. This signal contains detailed information about a stimulus, including local features and their global relationship. Indeed, experimental evidence is available which shows that cortical neurons can produce feature-specific phase lags

in their activity [44]. Theoretical studies argue that synchronous and dispersed activity are different functional modes of the same basic network structure [20]. The transition between these modes depends on the transmission delays in the lateral coupling. Moreover, Maass [48] has shown that neurons which encode information in their spike-timing have interesting computational properties. Therefore, we believe that the temporal population code provides a promising approach towards both, invariant pattern recognition and the understanding of the encoding of information by the nervous system.

Chapter 3

The enhanced temporal population code

3.1 Introduction

In the previous chapter, the basic principles of the temporal population code and some of its properties have been investigated. Here, we would like to discuss an enhanced version of the code. This code arises from a network whose units are characterized by their orientation and spatial frequency selective responses similar to neurons in primary visual cortex. In addition, the lateral connectivity is no longer isotropic but depends on the neural response preferences. In particular, long-range horizontal connections are preferentially linking neurons with co-oriented and co-axially aligned receptive fields [15]. These changes to the network have two major implications regarding its coding properties. First of all, the temporal population code becomes multi-dimensional because each population of a particular neuron type is treated as an individual channel carrying a single temporal population code. Second, due to the separation of different neuron types and the non-isotropic lateral connectivity, the resulting temporal population code is no longer stimulus rotation invariant. All together, this enhanced temporal population code provides more coding space and is therefore better suited for the reliable classification of large stimulus sets of increased complexity.

We show, that the enhanced temporal population code has similar basic coding properties as the model presented in the previous chapter. Further-

more, we demonstrate that the stimulus encoding is invariant to small deformations and preserves an intuitive notion of visual similarity. On the MNIST database [45], a widely used benchmark consisting of many hand-written samples of the 10 digits, it achieves nearly 95% of correct classifications. Thus, the encoding process performed by the network discards part of the information for the generation of invariant representations, while preserving the relevant information for the classification of stimuli. In order to have access to hundreds of stimulus classes, we generate a set of synthetic stimuli. These are used to investigate the scaling performance of the network, as well as the robustness and dimensionality of the resulting representation.

3.2 Methods

3.2.1 Network

As opposed to the model presented in the previous chapter, the smallest unit of the network is not a single neuron, but rather a column of neurons. These columns, however, are modeled as a single leaky integrate-and-fire unit with graded output. The time course of its membrane voltage $V(t)$ is described by the differential equation:

$$C_m \frac{dV}{dt} = -(I_{exc}(t) + I_K(t) + I_{leak}(t)) \quad (3.1)$$

where C_m is the membrane capacitance ($C_m = 0.2$ nF), and I represents the transmembrane current, i.e. excitatory input (I_{exc}), spike-triggered potassium current (I_K) and leak current (I_{leak}). These currents are computed by multiplying a conductance g with the driving force: $I(t) = g(t)(V(t) - V^{rev})$ where V^{rev} is the reversal potential of the conductance ($V_{exc}^{rev} = 60$ mV, $V_K^{rev} = -90$ mV, $V_{leak}^{rev} = -70$ mV). The column's activity at time t is given by $A(t) = a \cdot \Theta(V(t) - \theta)$ where $a \in [0, 1]$ is the activation of the column as determined by the applied input stimulus and the column's orientation and spatial frequency selectivity. Θ is the Heaviside function and θ is the firing threshold ($\theta = -55$ mV). Each time a column emits a spike, the potential is reset to $V_{rest} = V_{leak}^{rev}$. The constant leak conductance g_{leak} is 20 nS. The time course of the potassium conductance is given by

$\tau_K dg_K/dt = -(g_K(t) - g_K^{peak} \hat{A}(t))$ where $\hat{A}(t) = H(V(t) - \theta)$ with a time constant τ_K and a peak conductance g_K^{peak} ($\tau_K = 40$ ms, $g_K^{peak} = 200$ nS). The excitatory input to a column consists of two components. First, a constant driving conductance of 5 nS in conjunction with the time course of the membrane potential as given above yields a firing rate of approximately 42 Hz after frequency adaptation. Second, the synaptic conductances of the lateral connections between different columns.

Each column is characterized by its orientation and spatial frequency selectivity. Thus the columns can be parameterized by a triplet (\vec{x}, ϕ, ν) , where $\vec{x} \in [0, 1]^2$ is a two-dimensional vector specifying the center of the column's receptive field within the visual space, $\phi \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ is the column's preferred orientation and $\nu \in \{\text{high, medium, low}\}$ its preferred spatial frequency. The total number of columns is 8400, with an equal amount selective for each of the four orientations. The ratio of columns selective for the three different spatial frequencies is given by *high* : *medium* : *low* = 16 : 4 : 1. The columns selective for a particular spatial frequency are arranged in a regular grid spanning the $[0, 1]^2$ plane, where in turn four columns representing the different orientations share the same receptive field center.

The input to this network consists of a retinal grey-scale image of 80×80 pixels corresponding to the $[0, 1]^2$ plane. This image passes through an edge detection stage, by convolving it with a difference of Gaussian (DOG) kernel k_{ij} given by

$$k_{ij} = e^{-16r^2} - 1/4e^{-4r^2} \quad \text{with } r = \frac{\sqrt{i^2+j^2}}{3} \text{ for } i, j \in \{-3, \dots, 3\}. \quad (3.2)$$

The resulting contour is cropped to the original size of the image and represents the activity in the LGN, which serves as the input to V1 (fig. 3.2.1). Given the position $\vec{y} \in [0, 1]^2$ of the thalamic neurons within the visual space, the activation of a V1 column $a_{(\vec{x}, \phi, \nu)}$ is defined by the absolute value of a complex sum

$$a_{(\vec{x}, \phi, \nu)} = \left\| \sum_{\|\vec{y}-\vec{x}\| < r_{max}^\nu} e^{-\left(\frac{2\|\vec{y}-\vec{x}\|}{r_{max}^\nu}\right)^2} \times e^{i\frac{\vec{\phi}(\vec{y}-\vec{x})3\pi}{r_{max}^\nu}} \right\| \quad (3.3)$$

which essentially is the convolution of the input image with complex, oriented Gabor filters of different orientation and spatial frequency selectivi-

ties. $\vec{\phi}$ is the unit vector with angle ϕ . r_{max}^ν is the normalized radius of the circular receptive field of the V1 cell within LGN which is 0.05, 0.1 and 0.2 for high, medium and low spatial frequency selectivities respectively.

The lateral connectivity between the V1 columns is exclusively excitatory and characterized by (1) symmetric short-range and (2) asymmetric long-range connections (fig. 3.2.1). A cortical column a connects to a cortical column b if $\nu_a = \nu_b \wedge \vec{x}_a \neq \vec{x}_b$ and one of the following conditions are met:

1. $\phi_a \neq \phi_b \quad \wedge \quad$ “a and b are nearest neighbors”
2. $\phi_a = \phi_b \quad \wedge \quad |\angle(\vec{x}_b - \vec{x}_a, \vec{\phi}_a)| < 30^\circ \quad \wedge \quad \|\vec{x}_b - \vec{x}_a\| < l_{max}^\nu$

where l_{max}^ν is the maximal length for long-range connections, 0.35, 0.55, 0.75 for high, medium and low frequency selectivities respectively. The second condition describes the connectivity between columns with identical orientation selectivity, i.e. each column receives connections from columns within a sector with an apex angle of 30° , a radius of l_{max}^ν and the apex centered on the receiving column. The synapses are of equal strength w and are modeled as instantaneous excitatory conductances, while transmission delays $\tau_{a \rightarrow b}$ are proportional to $\|\vec{x}_b - \vec{x}_a\|$ with 1 ms/cell. Thus the total lateral synaptic input conductance at the post-synaptic column b is given by

$$g_b(t) = w \sum_{a \in C_b} A_a(t - \tau_{a \rightarrow b}) \quad (3.4)$$

where C_b is the pool of columns, from which b receives inputs, according to the conditions above.

In the discrete-time simulations, all the equations are integrated with Euler’s method using a temporal resolution $\Delta t = 1\text{ms}$.

3.2.2 Statistics

For the statistical analysis of the network’s response, the activities of the cortical columns are recorded over 100 ms after stimulus onset, yielding the vectors $\vec{A}_{\vec{x}\phi\nu} = \{A_{\vec{x}\phi\nu}(t) | t = 1, \dots, 100 \text{ ms}\}$. The temporal population response of the (ϕ, ν) -columns is then given by $\vec{A}_{\phi\nu} = \sum_{\vec{x}} \vec{A}_{\vec{x}\phi\nu}$. Thus the response of the network to a stimulus s is given by the twelve vectors

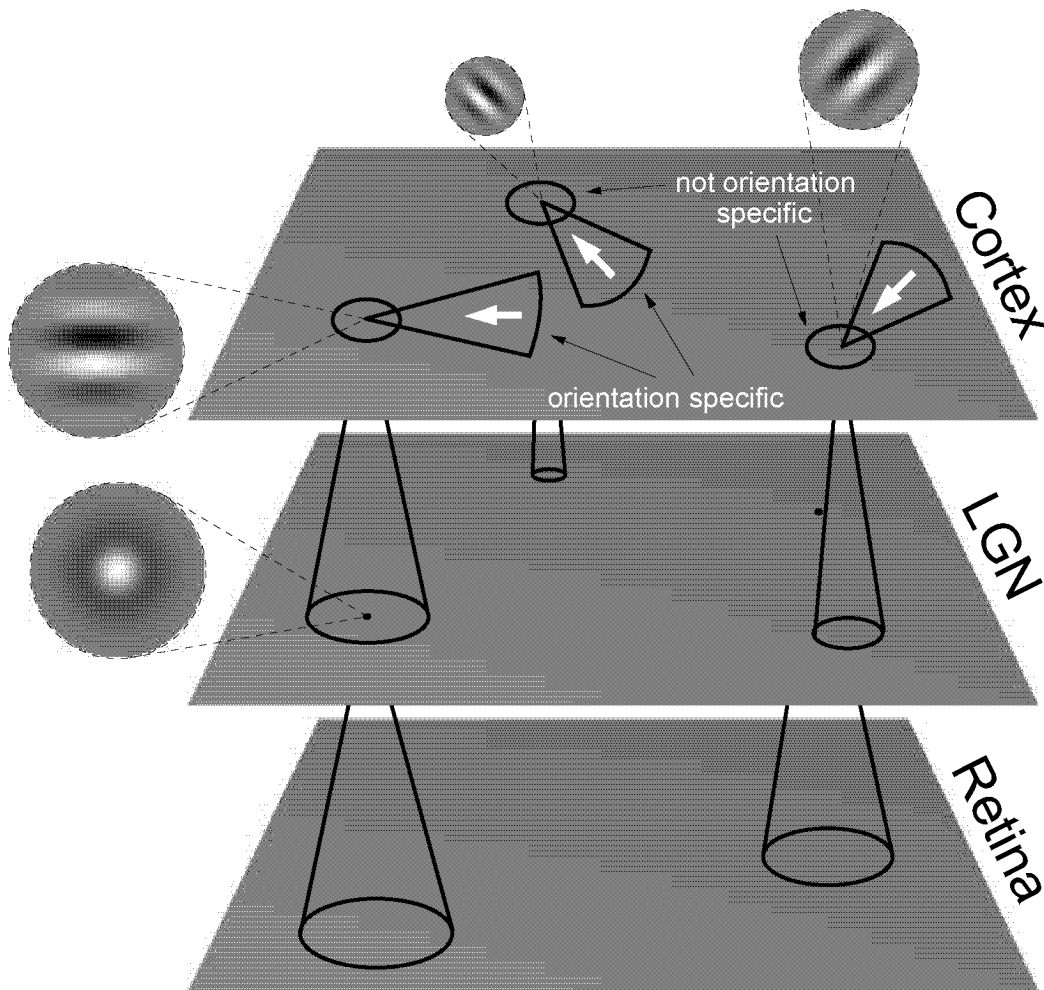


Figure 3.1: Schema of the network system. The input, present at the retina first passes through an edge detection stage, implemented by means of LGN cells with center surround receptive fields. The resulting contour of the stimulus in the LGN is then passed to the cortical columns, which in turn have different spatial frequency and orientation selectivities. The receptive fields of the columns are oriented Gabor patches of different sizes (spatial frequencies). The active columns within the cortical layer interact through lateral coupling, which is specific (selective) for the relative spatial frequency and orientation selectivities of the interacting columns (see text). This interaction leads to shifts of the spike timing of columns, such that the spatial stimulus presented at the retina is encoded in the temporal population activity of the cortical network.

$\{\vec{A}_{\phi\nu}(s)|\forall\phi,\nu\}$. Each vector is normalized to a maximum of one in order to remove any information in the size of the response about the stimulus' orientation and spatial frequency statistics. This conforms with the objective of the present study to investigate the amount of information encoded in the temporal domain only. We evaluate the encoding performance of our model by dividing the stimulus samples into test and training set. The training set is used to define prototypes, that are used for classification of the responses of the training samples. The performance of the network in encoding the stimuli was measured by clustering the set of temporal population responses for the stimuli into as many response classes R_β as there are stimulus classes S_α . A response class is characterized by the prototype response $\vec{A}_{\phi\nu}^\beta$, the averaged temporal population response of the network to the training samples of the corresponding stimulus class β . A test sample $s \in S_\alpha$ is assigned to the response class R_β to which it is closest. As a distance measure we use the sum over the Euclidean distances between the twelve corresponding vectors, i.e.

$$d(s, R_\beta) = \sum_{\phi,\nu} \|\vec{A}_{\phi\nu}(s) - \vec{A}_{\phi\nu}^\beta\|. \quad (3.5)$$

As in the previous chapter, the clustering is summarized by a hit-matrix N , where the components $N(S_\alpha, R_\beta)$ correspond to the number of times a stimulus sample from class α was assigned to the response class β (see sec. 2.2.2).

3.2.3 Stimuli

Synthetic stimulus classes are generated by choosing 5 random points \vec{p}_i within a circular disc with radius 0.35 centered in the input space given by the $[0, 1]^2$ plane. Each pair of points is connected by a line with a probability of 0.3 (fig. 3.2). The points are jittered following a random distribution N with variance σ_{pos}^2 for all the samples of a stimulus class: $\vec{p}'_i = \vec{p}_i + \delta\vec{p}_i$ for $i \in \{1, \dots, 5\}$, where $\delta\vec{p}_i \in [N(0, \sigma_{pos}^2)]^2$. The thickness of the bar d_{ij} connecting \vec{p}'_i and \vec{p}'_j is drawn from a normal distribution, i.e. $d_{ij} \in N(d, \sigma_{width}^2)$ with $d=0.12$ (see fig. 3.2). From a given set of stimulus classes, three sets of samples are generated, each with a different within class variability, i.e. $\sigma_{pos} = 0.03, 0.04, 0.05$ and $\sigma_{width} = 0.021, 0.025, 0.029$













Class	low variability	medium variability	high variability
			
			
			

Figure 3.2: Three examples of the synthetic stimulus classes with 4 samples each for the three different levels of variability. A stimulus class is defined by a set of random points $\{\vec{p}_i | i = 1, \dots, 5\}$ and a randomly chosen pairwise connectivity pattern between the points. The samples to a stimulus class are generated by jittering the position of the individual points as well as varying the thickness of the bars between the points to various degrees.

respectively.

3.3 Results

In the following we perform several experiments, in order to investigate the coding properties of the proposed enhanced network. First we provide examples of the network responses. Thereafter, a number of controls are presented investigating the coding performance and stability with respect to several network parameters. Subsequently, the properties of this code and in particular the dimensionality of the coding space is analyzed. Finally, we test the scaling properties of the temporal population code with respect to the number of stimulus classes presented.

3.3.1 Example of stimulus classification

First, we would like to provide a first insight into the temporal population code generated by the proposed network and introduce some basic terminology and performance measures. The temporal population response of the network consists of 12 activity traces, one for each neuron type, defined by their orientation and spatial frequency selectivity. Example traces for

a	0				
b	1	0			
c	1.83	1.69	0		
d	1.75	1.52	1.42	0	
e	1.51	1.43	1.6	1.66	0
	a	b	c	d	e

Table 3.1: Mutual Euclidean distances between the different samples shown in figure 3.3, normalized to the distance between the samples (a) and (b) which belong to the same stimulus class.

the four orientations and the highest spatial frequency are shown in figure 3.3a-e for different stimuli. The population response shows synchronous bursts of activity over a coarse time-scale combined with dispersion on the time-scale of individual bursts. This dispersion is due to the interaction of active neurons through the delayed signaling over their lateral coupling. Due to the spike triggered potassium currents, we observe frequency adaptation in the form of a prolongation of the inter-spike intervals. Stimuli *a* and *b* are samples from the same stimulus class, while *c* - *e* are from different classes. The mutual Euclidean distances between the five samples in fig. 3.3 are given in table 3.1. The distances are normalized to the distance between the two samples *a* and *b* that are from the same stimulus class. All normalized Euclidean distances between samples of different classes lie above one.

3.3.2 Robustness of encoding

In the following experiments, we investigate to what extent some of the properties found in the previous chapters also hold for the enhanced network. We show the dependency of classification performance on the synaptic strength in the lateral coupling, which is the key ingredient to the present approach. Furthermore, we survey coding stability with respect to synaptic noise and with respect to stimulus variability by testing the proposed coding scheme in a standard benchmark for pattern recognition, the MNIST database. For these more quantitative investigations, the network responses to stimulus sets will be clustered (see methods), in order

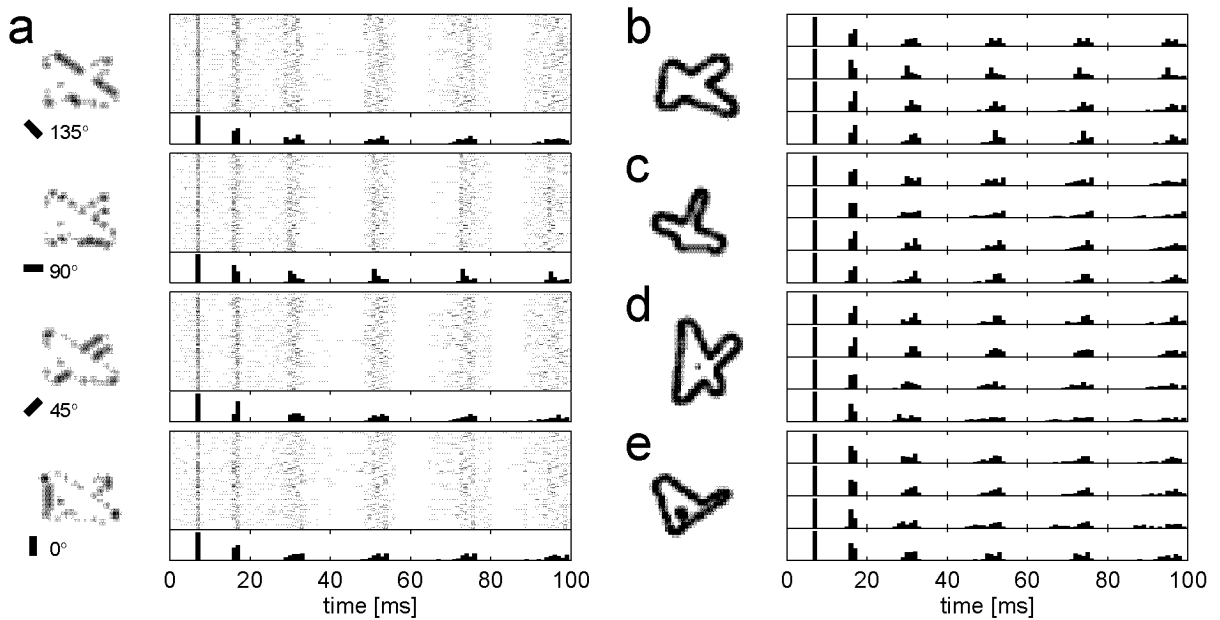


Figure 3.3: **(a)** Activity of the columns with highest spatial frequency selectivities and different orientation preferences (0° , 45° , 90° , 135°) for a stimulus (left), along with the spike raster plot and the corresponding activity histograms over time (right). **(b)-(e)** Activity histograms over time of the four different populations of orientation selective columns for samples of the same (b) and different (c)-(e) stimulus classes than in (a). Note that in (a), the filtered stimuli are shown individually for the different orientations. In (b)-(e), the four orientations are superimposed.

to determine the information available in the temporal population code. Naturally, this information depends on the clustering algorithm used and therefore we will compare different approaches.

The objective of the first experiment is to determine the dependency of the networks encoding performance with respect to the lateral synaptic strength w . In particular, the contribution of the populations of cortical neurons selective for the three different spatial frequencies is analyzed. The stimulus set consists of 50 stimulus classes with low variability, 100 samples each, where 50 samples are used to produce the prototype responses (“training set”) and 50 samples are clustered (“test set”). The simulation was performed for $w \in [0, 1]$ nS. In order to determine the contribution of the different frequency selectivities, the clustering was performed based on the population responses for the different spatial frequencies individually. In fig. 3.4a the percentage of correctly classified stimuli is plotted as function of w for the three spatial frequencies. For $w = 0$ nS, the performance is around chance level for all frequencies, because there are no effective lateral connections, which are necessary for transforming the spatial stimulus structure into the temporal domain. For increasing lateral synaptic strength, however, the performance increases in all three cases, with an optimal performance for $w_{opt}^\nu = 0.92, 0.32, 0.2$ nS for high, medium and low spatial frequency selectivities respectively. In all cases, performance remains stable at a high level over a wide range of synaptic strength.

Performance in terms of correctly classified samples does not depend on the type of errors made by the clustering. Mutual information, however, does account for the nature of errors. This can be clarified by investigating two idealized cases [72]. In the first case, stimulus classes group naturally into *super-classes* of size Z , whereas stimulus classes from different super-classes are perfectly discriminated and stimulus classes within the same super-class are not discriminated at all. In this case, the percentage of correctly classified samples is $1/Z$ and the mutual information is equal to $\log_2(C/Z)$, where C is the number of stimulus classes. In a second case, we assume that no similarity structure exists among the stimuli, such that they are discriminated with probability q or confused with any of all the others with probability $1 - q$. In this case the percentage correct is $q + (1 - q)/C$ and mutual information is given by

$$\frac{1 + q(C - 1)}{C} \log_2(Cq + 1 - q) + \frac{(C - 1)(1 - q)}{C} \log_2(1 - q). \quad (3.6)$$

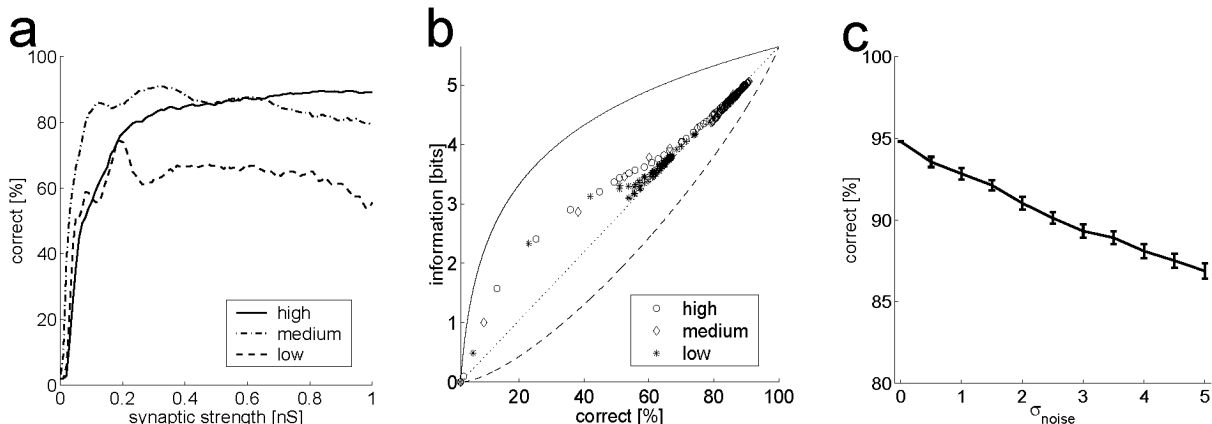


Figure 3.4: **(a)** The percentage of correctly classified stimuli as a function of the lateral synaptic strength. The clustering is based on the the network responses of three populations of neurons defined by their spatial frequency selectivity individually. **(b)** Information vs. Percentage Correct. The arrangement of the data points from (a) relative to the two sample cases (see text). **(c)** Influence of synaptic noise on encoding. Percent of correct classifications as a function synaptic noise σ (mean \pm std, $n=50$).

These relationships are illustrated in fig. 3.4b (with $C = 50$), which shows mutual information versus the percentage of correct classifications for the two sample cases described above. Plotting the data from fig. 3.4a in this graph reveals that for optimal performance, there is a nearly linear relationship between the two performance measures. The fact that the experimental relationship is between the two extreme cases indicates that the encoding of stimulus similarity is slightly more complex. However, we can not draw more specific conclusions, since an arbitrary number of different similarity structures generate similar relationships. An example, in which a linear relationship exists between the information available and the percentage correct (close to what the data show), is when super-classes exist but also each stimulus can be discriminated, with a certain probability, within its super-class [72]. Regarding code dimensionality, we can make the following observation: if all pairs of stimulus classes are equally likely to be confused (dashed line), the stimulus classes have roughly the same distance from each other. This can only take place in a high-dimensional coding space. The other case (forming super-classes, solid line) can exist

in a low-dimensional coding space. However, although the data points in fig. 3.4b appear to lie closer to the high-dimensional case no quantitative statement can be made, because of the lack of a metric within the representation in fig. 3.4b. The question about code dimensionality will be pursued further below.

As shown above, the performance of the network in encoding the stimuli does not critically depend on the exact strength of the lateral coupling, as soon as a threshold value of w is exceeded. The objective of the next experiment is to determine, whether this finding also holds for dynamically changing synaptic strengths due to the putative highly unreliable transmission of signals across synapses [6]. Synaptic noise is modeled by perturbing the individual synaptic conductances dynamically; each conductance is multiplied by a random factor f , drawn from a normal distribution with mean one and variance σ^2 , i.e. $f \in N(1, \sigma^2)$. For $f < 0$ the synaptic conductance is set to zero, which corresponds to a synaptic failure. For increasing noise the system's performance in encoding the input stimuli decreases approximately linearly (fig. 3.4c). The number of correctly classified stimuli decreases by not more than 3% for $\sigma = 1$, which corresponds to a signal to noise ratio of 1. Thus the encoding of the stimuli by the enhanced network is similarly stable with respect to synaptic noise as the simpler network presented in the previous chapter.

A different and important type of noise source in pattern recognition, is the pattern variability within stimulus classes. Therefore, the network is tested on handwritten digits from 250 writers contained in the MNIST database [45] (fig. 3.5a), which is a standard benchmark in the domain of character recognition. To facilitate comparison with other systems tested on this database we used the original partitioning into a training-set and a test set with of 5400 samples and 950 samples per digit respectively. Our model is capable of classifying 94.8% of the stimuli correctly (fig. 3.5b).

So far, the clustering of the network responses was performed by comparing the test samples to the prototype responses of each class, which are essentially the average responses of the network to the training samples of a class. Here, we investigate the *k-nearest-neighbors* (KNN) clustering method. This algorithm determines the pairwise Euclidean distance between the sample which is about to be classified, and all the other samples, from the same and all other classes. The k nearest neighbors are then determined, and the sample is assigned a classification based on the

0 0 0 0 0
1 1 1 1 1
2 2 2 2 2
3 3 3 3 3
4 4 4 4 4
5 5 5 5 5
6 6 6 6 6
7 7 7 7 7
8 8 8 8 8
9 9 9 9 9

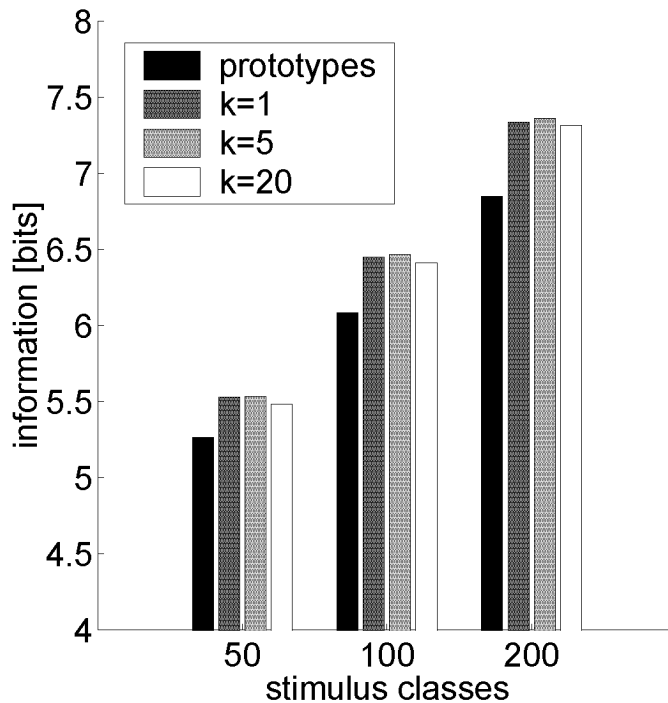


Figure 3.6: Clustering methods. Mutual information for three stimulus sets with 50,100 and 200 stimulus classes. Two different types of clustering methods have been used: prototype clustering and KNN clustering for $k = 1, 5, 20$.

prevailing classification of these k neighbors. Fig. 3.6 shows the mutual information for stimulus sets consisting of 50, 100 and 200 stimulus classes based on the prototype clustering as well as for the KNN clustering with $k = 1, 5, 20$. While there is only a small performance difference between the KNN clusterings, they outperform prototype clustering for all three stimulus sets. Thus using the prototype clustering as the method of choice is save with respect to performance over-estimation as well as computationally more efficient. In addition, the absolute performance difference between KNN and prototype clustering is rather small, and therefore it is meaningful, to analyze the topology of the cluster centers in the potentially high-dimensional space.

3.3.3 Code dimensionality

By comparing the prototype responses for different stimulus classes, we can estimate their similarity. In fig. 3.7 the dendrograms for two different sets of stimuli are shown with (a) similar and (b) different local features. The height of the vertical bars between stimulus classes/superclasses correspond to their (averaged) mutual Euclidean distances, normalized to the mean distance of the samples to their corresponding prototype. The pairs of stimulus classes in fig. 3.7a with a distance of approximately one are visually very similar. The mutual distances between the stimulus classes in fig. 3.7b are all significantly above one, however, they have all approximately the same distance from each other, i.e. suggesting a high-dimensional coding space.

A more systematic overview on the distances between samples of the same and different classes is given in fig. 3.7c, for 100 stimulus classes, 250 samples each. The distances are normalized with the mean distances of the samples to their own prototype, and this distribution is represented by the histogram termed *internal*. The histogram termed *external* shows the distance distribution of samples to other prototypes than their own. The distances between stimulus classes vary within a limited range only. This is the hallmark of a high dimensional representation. Please note, however, that we are considering a finite number of stimulus classes only. As a Gedankenexperiment N stimulus classes can be distributed over the corners of a $\log_2(N)$ dimensional hyper cube. This would equally lead to a narrow distribution of distances between samples and all prototypes. Thus using this type of reasoning we can not make statements on dimensionalities higher than $\log_2(N)$. For the present work this amounts to roughly 10 dimensions.

As a further control, the *random internal* and *random external* histograms show the distance distributions of samples to randomly generated prototypes. These prototypes are defined as random patterns that give rise to the same activity statistics as the structured stimuli within populations of neurons with the same spatial frequency selectivities. In addition, the distribution called *random internal*, also preserves the activity statistics within populations of neurons with the same orientation selectivities. In both cases, however, the distances lie significantly outside of the internal distribution. Hence, with a reasonable choice of a classification thresh-

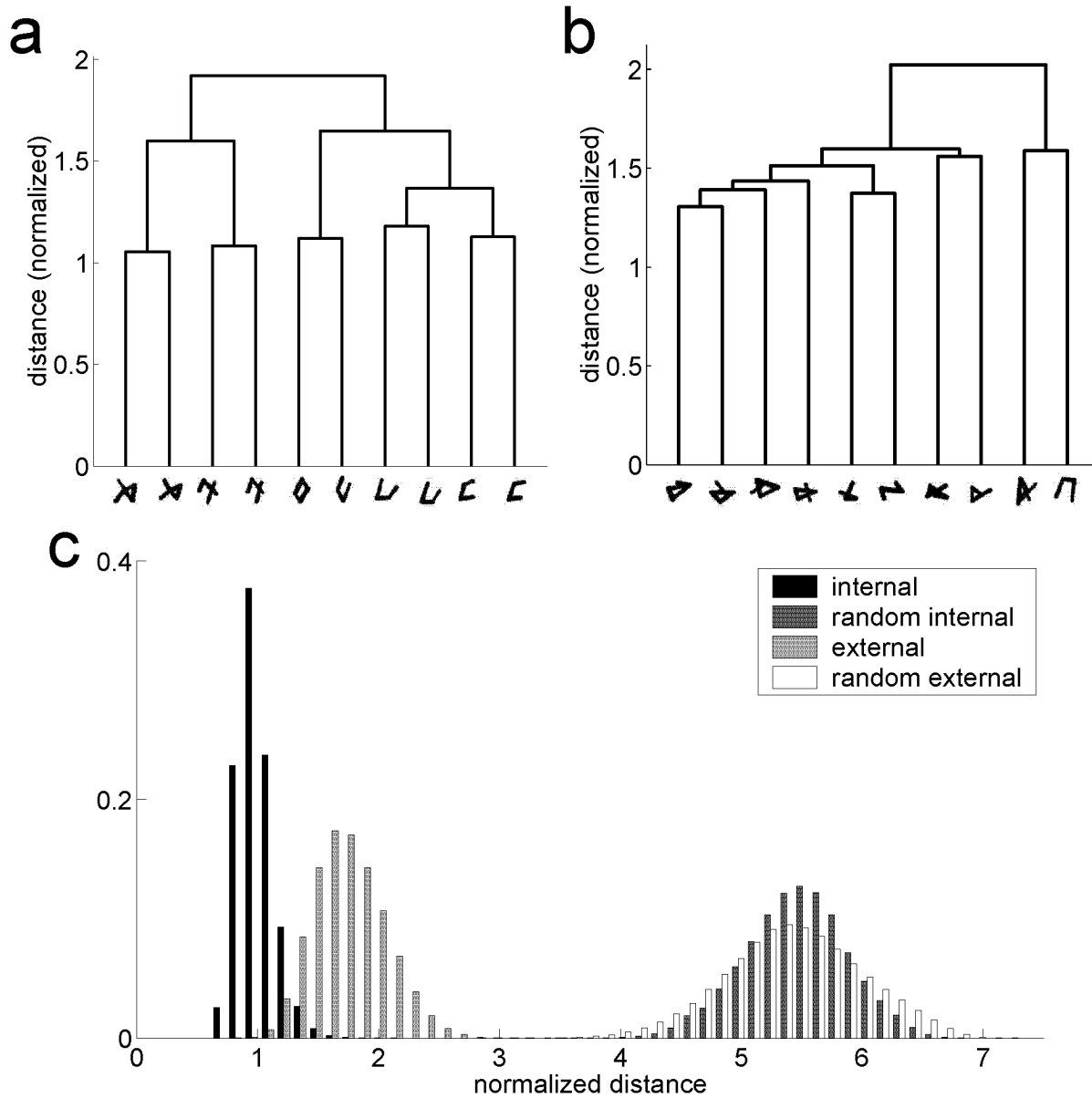


Figure 3.7: **(a)** & **(b)** Dendrograms for 10 stimulus classes with (a) similar and (b) different local features. The height of the vertical bars connecting single or groups of stimulus classes corresponds to the their (averaged) mutual distance. All distances are normalized to the average distance of samples to their own prototypes. **(c)** Distance distributions of samples to their own (internal) and other (external) stimulus prototypes, as well as to random prototypes with the same (random internal) and other (random external) orientation statistics.

old the number of false positives is virtually zero. Thus, it is unlikely that in the proposed encoding scheme random stimuli are confused with structured stimuli. The two distributions for the random prototypes have approximately the same mean and only differ by their variances, the external being slightly broader than the internal. This can be explained by the external prototypes having more degrees of freedom which leads to a broader distribution. More importantly, however, this shows that the orientation statistics of a stimulus does not contribute significantly to the information encoded in the temporal population code. Rather the spatially structured topological properties of a stimulus are encoded.

So far, our results suggest a high-dimensional coding space. A traditional tool to determine the dimensionality of a data set in a more rigorous way is *principal component analysis* (PCA) [63]. PCA involves a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible.

We perform PCA on the network responses to a stimulus set of 800 classes, 250 samples each. Here, we only consider the population of cells selective for the highest spatial frequency. A PCA basis is computed from all network responses, irrespective of their orientation tuning. The network responses to a stimulus are then projected upon this basis yielding a four-dimensional PCA coefficient for each component and stimulus, i.e. one dimension for each orientation. Clustering is then performed using Euclidean distance between corresponding coefficient-vectors as a similarity measure. Fig. 3.8 shows the additional and individual information contributed by each component (4 dimensions each, dashed and solid lines respectively), starting with the first order component in the former case. Higher order components carry a significant amount of information individually, however, this information is partly or fully redundant with information provided by lower order components. The five first components account for 91% of the total information encoded in the network response. Thus, the dimensionality of the code generated by the network is approximately 20.

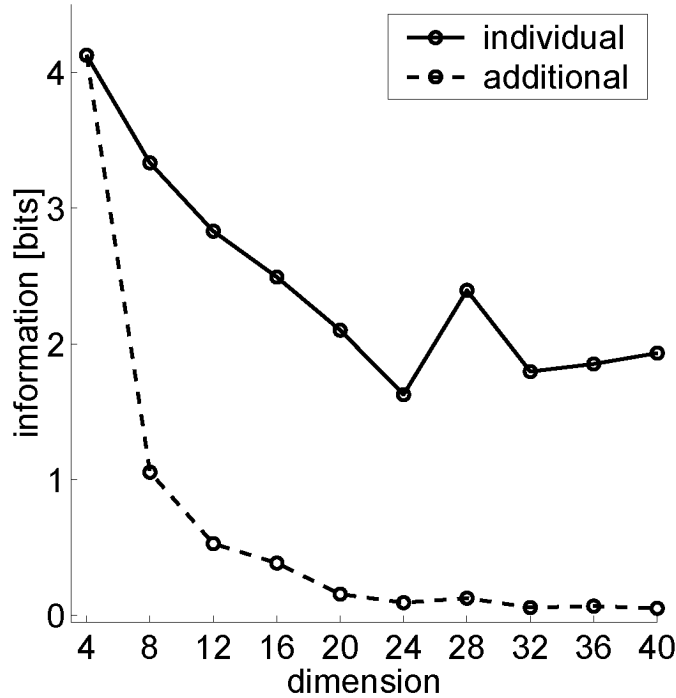


Figure 3.8: Information as a function of the number of PCA components/dimensions. The additional information provided by subsequent components (data points), each comprising 4 dimensions, is shown by the dashed line. The information encoded the individual principal components individually is given by the solid line.

3.3.4 Scaling with the number of stimulus classes

The robust encoding suggests a favorable scaling to a large number of stimulus classes. Therefore, in this last control, we investigate the scaling properties of the model presenting stimulus sets to the network with 50, 100, 200, 400, 800 and 1600 classes, 250 samples each, for three different stimulus variabilities. The prototype responses are generated from 50 samples of each stimulus class. The remaining 200 samples are clustered. The network’s performance is shown in fig. 3.9a, where the encoded information is plotted as a function of the number of stimulus classes. Optimal performance ($I = \log_2(\#classes)$) is indicated by the thin dashed line. For an increasing number of stimulus classes the encoded information stays close to the theoretical limit. Thus, the proposed encoding scheme scales to problems of interesting complexity.

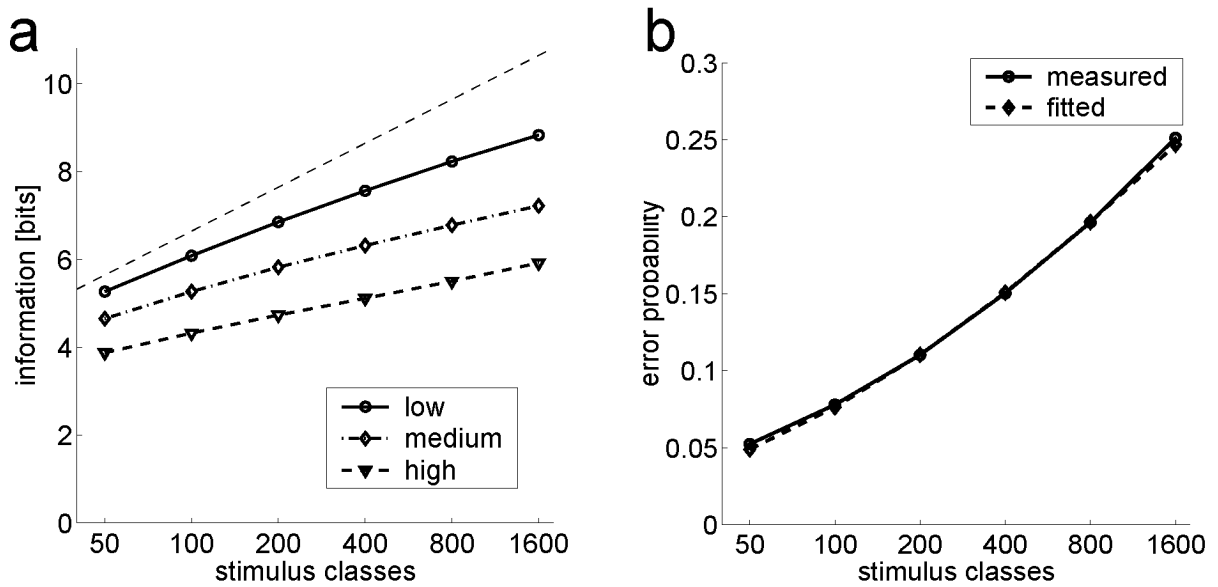


Figure 3.9: **(a)** Information encoded by the network as a function of the number of stimulus classes. For classification, prototype clustering has been used for the three different levels of stimulus within class variabilities. Optimal performance ($\text{Information}=\log_2(\#classes)$) is indicated by the thin dashed line. **(b)** Classification error probability as function of the number of stimulus classes derived from the experimental data (solid line) and fitted (dashed line, see text) for the lowest level of within class variability.

In the following we would like to analyze the error probability as a function of the number of stimulus classes in more detail. Given the distribution functions of distances between samples and their corresponding prototypes $\phi_{int}(x)$ and between samples and non-corresponding prototypes $\phi_{ext}(x)$, where x designates the distance, the error probability as a function of the number of stimulus classes N is given by

$$p_{err}(N) = \int_{\alpha(N)}^{\infty} \phi_{int}(x) dx \quad (3.7)$$

where $\alpha(N)$ is the expectation of the minimal distance of a sample to any of N non-corresponding prototypes given by

$$\alpha(N) = \int_{-\infty}^{\infty} xN(1 - \Phi_{ext}(x))^{N-1} \phi_{ext}(x) dx \quad (3.8)$$

where $\Phi_{ext}(x) = \int_{-\infty}^x \phi_{ext}(\hat{x}) d\hat{x}$ is the cumulative distribution function of $\phi_{ext}(x)$. If we assume $\phi_{int}(x)$ and $\phi_{ext}(x)$ to be Gaussian, $p_{err}(n)$ only depends on the widths of these distributions, i.e. σ_{int} and σ_{ext} , and their relative distance $d\mu = \mu_{ext} - \mu_{int}$. From the experimental data, we measure $d\mu = 0.76$, $\sigma_{int} = 0.15$ and $\sigma_{ext} = 0.3$. Fig. 3.9b shows the measured error probability as a function of stimulus classes (solid line) together with the fitted theoretical error probability $p_{err}(N)$ (dashed line) as given above, with $d\mu$ fixed to the experimental value of 0.76. The resulting standard deviations are $\sigma_{int} = 0.21$ and $\sigma_{ext} = 0.18$. This discrepancy between experimental and fitted values arises from the assumption that $\phi_{int}(x)$ and $\phi_{ext}(x)$ have a Gaussian distribution, which is obviously a simplification (fig. 3.7c). Nevertheless, the simple heuristics outlined above give a surprisingly good overall description of the network's scaling properties for large stimulus sets. In addition it gives us a tool at hand to extrapolate network performance to even larger stimulus sets.

3.4 Discussion

This chapter introduced an enhanced network for the generation of a temporal population code, which consists of orientation and spatial frequency selective units. Using a set of artificially generated visual stimuli, we show

that this temporal population code favorably scales up to hundreds of stimulus classes. This not only holds for k-nearest neighbor clustering, but also for the computationally more efficient prototype clustering. Moreover, we show that this encoding is robust with respect to variations in synaptic strength and noise in the lateral connections as well as stimulus variability.

In a widely used benchmark on the recognition of hand written digits nearly 95% correct classification was achieved. This falls short of the performance of the very best specialized character recognition systems [45]. However, it compares well to an approach, in which a k-nearest neighbor clustering is applied directly to the spatial representation of the stimuli centered at a common position [45]. Thus, the encoding process performed by the network discards part of the information for the generation of invariant representations, while preserving the relevant information for classification of the stimuli.

One of our aims was to assess the dimensionality of the encoding. Several measures presented argue that it is high-dimensional. Comparing the percentage correct with information encoded in the temporal population code with the two extreme cases of either a high- or a low-dimensional code shows that our model does not match either of these two extremes. However, the histogram of the distances between different stimulus classes reveals a very narrow distribution which argues for the high-dimensional case; it is not possible to arrange a large amount of stimulus classes within a low-dimensional space while keeping the distance distribution narrow. Moreover, the principal component analysis shows that the first 5 components, i.e. 20 dimensions, suffice to extract most of the information available in the temporal population code. Recently, a model was proposed that maps continuous time inputs onto a high-dimensional spatial representation [49]. Hence, it is of interest to investigate in future work, whether this model could be used for the classification of a high-dimensional temporal population code.

A number of studies propose a role for temporal dynamics in the formation of visual representations. One of the first proposals [66] suggests that local features of visual stimuli are encoded in the temporal patterns of spikes of single neurons. This concept has been investigated in several stages of the hierarchy of the visual system, including LGN, V1 and inferotemporal cortex [52, 66, 67]. Based on principal component analysis of the response patterns of single neurons in repeated recordings, a significant

amount of information carried by the first (\cong firing rate) as well as the higher components (\cong temporal pattern) was found [90]. An important difference with the present study is that here we consider the encoding of stimuli by populations of neurons in a single run, rather than by a single neuron in repeated runs [30]. This also explains why the peri stimulus time histograms between these two cases appear very different, being smooth in the former and having high variability in the latter case. We show that this variability provides an efficient coding substrate. Moreover, in related work it is shown that it also has a profound influence on the interactions within a network [13]. Surprisingly, to our knowledge, no experimental data are available on the variability over time of the activity of whole populations of neurons.

Temporal coding on a millisecond time-scale is observed in multiple modalities. For instance, in the auditory system, where also the stimuli themselves vary on this time-scale, Bialek and colleagues show that information about time dependent input can be conveyed by spike timing [68]. Evidence for a neuronal representation of auditory stimuli by spike timing has also been found in grasshoppers [50]. In the visual system, motion stimuli are temporally varying inputs, which for example can be reconstructed from the spike timing of the H1 neuron in the fly [21]. One of the earliest studies of the fast dynamics of neuronal systems was performed in the olfactory system [24]. Indeed, the most direct evidence for the behavioral relevance of temporal coding has recently been obtained in the olfactory system of the honeybee [51]. Moreover, this study also revealed that the olfactory system does transform relatively constant sensory inputs into a time-varying population response at the millisecond time scale. Hence, this suggests that temporal population coding arising out of the inherent dynamics of a neuronal system, as investigated here, is a valid hypothesis for the encoding mechanisms underlying sensory processing.

Chapter 4

Involving the motor system in decision making

The objective of the previous two chapters was to present a particular example of how temporal coding in combination with non-topological statistics can yield powerful computational mechanisms for the task of invariant pattern recognition. In this chapter, we would like to present a code which relies on non-topological statistics only without temporal coding. In addition, this code is employed at the motor system of a complete behavioral system. Thus, non-topological statistics are not only of interest for the interpretation of sensory information. Furthermore, we show the invariance properties of non-topological codes as discussed in the previous chapters also applies to more abstract concepts such as the invariant detection of behaviorally ambiguous situations.

4.1 Introduction

Traditionally, behavioral control is seen as emerging out of three stages: sensory processing, decision making and motor execution. Recently, however, physiological and behavioral studies have shown that the borders between sensory and motor systems become more and more vague [42, 56, 70, 75]. Furthermore, there is increasing evidence, that motor systems may not only be important in altering the physical state of an organism, but also

its cognitive state [28, 32]. Georgopoulos et al. have found direct evidence for task related modulation of activity in motor systems [19], which makes the latter a candidate not only for the execution of movements but also for their planning. Indeed, a recent theoretical study has shown that in order to satisfy constraints observed in optimal decision making in humans, the motor system should be seen as an integral component of a decision making process [92]. This integration of the motor system with decision making and perceptual stages raises the question whether the original tri-ate sensing-deciding-acting captures biological reality [70].

Involving the motor system in decision making has a number of advantages. First, decisions automatically adhere to the constraints given by the morphology of the behaving agent [70]. While the sensory input to an agent is in general task and goal unspecific, the motor output conforms to physical as well as task related boundary conditions. Thus, introducing category boundaries in processing of stimulus features at a motor level rather than at the sensory input appears to be more natural. Variations of stimuli which do not have a consequence for behavior can be clearly differentiated from those that do. This may be exploited to combine highly reproducible performance on particular local aspects of a task with a flexible behavioral repertoire on a more global scale. These properties motivate a further investigation of such an integrated decision system.

We use a neural model of sensorimotor integration to study the possible role of the motor system in interpreting behavioral situations. An autonomous mobile robot learns to perform a simple but still complex enough navigation task which requires different levels of behavioral control, i.e. navigation in a maze of lines. At a local level the agent has to learn to accurately follow a line, thus a very stereotyped and reproducible behavior. At a more global level however, the agent has to retain a flexible behavior in order to optimally explore the maze. Thus line following in a maze is a behavioral paradigm which allows to study the interactions and transitions between different possibly conflicting behavioral requirements. The robot employs a form of reinforcement learning [83], in order to learn the mapping from the sensory input provided by a camera to its motor output. We show, both in simulation and real-world experiments, that the robot's uncertainty at the motor level reflects the presence of multiple behavioral options. A simple non-topological measure of uncertainty at the motor level therefore serves to detect behaviorally relevant situa-

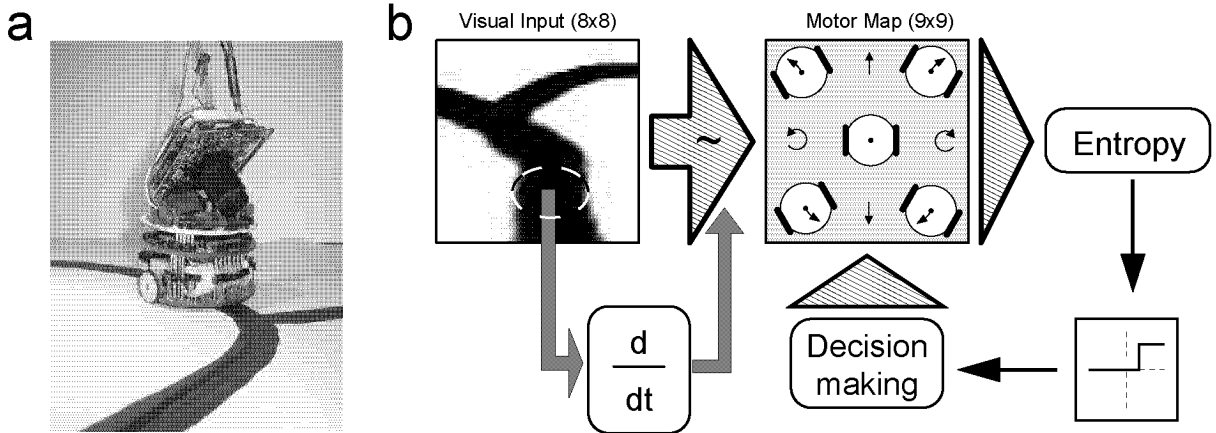


Figure 4.1: **(a)** The micro-robot *Khepera*, following a line drawn on the floor. A CCD camera is mounted on top of the cylindrical body with an inclination angle of $\approx 45^\circ$. **(b)** The network model. The visual input, with a resolution of 8×8 pixels is mapped onto the motor-map with 9×9 units, each of which corresponds to a pair of speeds for the left and right motors respectively. This sensorimotor mapping consists of parallel perceptrons which are learned by a form of reinforcement learning. Initially random actions are generated. These actions are validated by measuring the change in activity captured by the receptive field in the visual input (dashed circle): black lines moving in/out of the receptive field generate positive/negative learning signals. In parallel, the entropy of the motor-map determines the activation of the decision making mechanism. If the entropy surmounts a dynamic threshold (see text), random reflexes with equal probability are triggered causing the robot to bias its actions towards right or left turns.

tions. Thus, the motor system may not only change the physical state of an organism but also constitute an integral part of the decision making system.

4.2 Methods

Experiments were performed using both, the real-world robot *Khepera* (K-team, Lausanne, Switzerland, see fig. 4.1a) as well as a simulated virtual approximation of the latter programmed in C++ using OpenGL (sec. A.3.3). The robot has a circular body (radius = 2.25 cm) with two wheels attached

to its side each controlled by an individual motor. In the following we use the robot's body radius as our length unit. The visual input is provided by a camera mounted on top of the robot and inclined toward the floor with an angle of approximately 45° . The network model is simulated on a Linux computer using a neural network simulator programmed in C++ (see Appendix A). The update frequency of the simulation is 25 Hz which corresponds to the frame rate of the camera. On average, the robot moved with a speed of approximately 0.7 body radii per second.

The environment consists of a white rectangular arena surrounded by walls and a maze of black lines drawn on the floor (see fig. 4.2a). The virtual environment is a square with a side length of 35 body radii and contains 36 crossings of three lines each. Because the tethered real-world robot has a limited freedom of movement, the corresponding environment was designed accordingly smaller, i.e. 28×21 body radii² with 12 crossings only.

A schema of the model investigated in this study is shown in fig. 4.1b. The grey scale image of the floor in front of the robot with a resolution of 8×8 pixels constitutes the input to the system. The 9×9 units of the motor-map, which stand for a combination of left and right wheel speeds each, represent the output of the system. In the following we will discuss the different components of the model in detail.

4.2.1 Sensorimotor mapping

The mapping between the visual input and the motor output is responsible for the robot's controlled movement along lines. The general conditions for the acquisition and representation of this mapping are the following: (1) it can be learned based on a reinforcement type learning signal (2) the particular choice of learning system does not imply an a priori restriction on the type of possible mappings. The *parallel perceptron* [10] conforms to these requirements and is therefore used to learn the sensorimotor mapping. The visual input \mathbf{z} is processed by $n = 10$ single layer perceptrons whose individual outputs are given by $o_i = \mathbf{w}_i \cdot \mathbf{z}$ for $i = 1 \dots n$ where \mathbf{w}_i is the weight vector of perceptron i . The output p of the parallel perceptron is

then given by

$$p = \frac{1}{n} \sum_{i=1}^n \Theta(o_i) \quad (4.1)$$

where $\Theta(\cdot)$ is the Heaviside step-function, thus $p \in [0, 1]$.

The units in the robot's motor-map correspond to different robot movements as given by an instantaneous two-dimensional speed vector $\mathbf{s} = \langle s_L, s_R \rangle$ where s_L and s_R are the speeds of the left and right motors respectively. Each unit's activity a^s is controlled by the output p^s of an individual parallel perceptron, such that

$$a^s = f_\theta(p^s) \text{ where } f_\theta(x) = \begin{cases} 0 & x < \theta \\ x - \theta & x \geq \theta \end{cases} \quad (4.2)$$

The ensemble of motor units $\{a^s | \forall \mathbf{s}\}$ constitutes a stochastic motor-map, within which a particular unit $\hat{\mathbf{s}}$ has the probability $p^{\hat{\mathbf{s}}} = a^{\hat{\mathbf{s}}} / \sum_{\forall \mathbf{s}} a^s$ of being the winning unit. Due to the inertia of the real-world robot, executed motor actions are low-pass filtered, such that the actual motor-speeds are running averages over the past winning units. For consistency between real-world and virtual experiments, the motor-commands are low-pass filtered in order to smoothen the movements of the robot, i.e.

$$\langle \mathbf{s} \rangle_{t+1} = \frac{1}{\tau} \mathbf{s}(t) + \left(1 - \frac{1}{\tau}\right) \langle \mathbf{s} \rangle_t \quad (4.3)$$

where $\tau = 4$ simulation time-steps is a time-constant which approximately characterizes the inertial properties of the robot.

4.2.2 Learning

The learning signal γ^s for a parallel perceptron \mathbf{s} is given by the product of a non-specific performance measure α which is directly derived from the visual input, and a perceptron specific feedback β^s from the motor-map, i.e. $\gamma^s = \alpha \cdot \beta^s$.

The performance measure α is given by the temporal derivative of the summed activity from a small receptive field in the lower center of the visual input (see fig. 4.1b). This signal is positive/negative if a line moves in/out of the receptive field respectively and zero if the robot moves such that the receptive field stays centered on the line.

The perceptron specific factor is given by the Gaussian

$$\beta^s = \exp\left(-\frac{\|\langle \mathbf{s} \rangle_t - \mathbf{s}\|^2}{2\sigma^2}\right) \quad (4.4)$$

where $\langle \mathbf{s} \rangle_t$ is the actual motor-speeds as given in eq. 4.3, σ^2 is the variance and $\|\cdot\|$ is the Euclidean distance. Thus, β^s localizes learning to the motor units which lie close to the current motor-speed.

The weight vectors of the parallel perceptron \mathbf{s} are updated according to

$$\mathbf{w}_i^s(t+1) = \mathbf{w}_i^s(t) + \eta \cdot \Delta_i^s(t) \quad (4.5)$$

where η is the learning rate and

$$\Delta_i^s(t) = - (\|\mathbf{w}_i^s(t)\|^2 - 1) \cdot \mathbf{w}_i^s(t) + \begin{cases} \gamma^s(t) \cdot \mathbf{z}(t - \tau) & \text{for } \gamma^s(t) \cdot o_i(t - \tau) < 0 \\ 0 & \text{otherwise} \end{cases}$$

The first term moves $\|\mathbf{w}_i^s\|$ towards 1. The second term selectively updates the weight vectors for which $o_i = \mathbf{w}_i^s \cdot \mathbf{z}$ has opposite sign of γ^s . The time constant τ is given by the inertial properties of the robot (see eq. 4.3).

4.2.3 Decision making

The robot is equipped with three basic prewired reflexes: driving straight, turning to the right and turning to the left. These reflexes bias the motor-map rather than imposing particular motor-actions, by projecting linear activity ramps onto the motor-map, i.e. $a^s = f_\theta(o^s + b^s)$. The activity ramps b^s for the three reflexes mentioned above are defined as follows:

$$b^s = B + C \cdot \begin{cases} s_L + s_R & \text{driving straight} \\ s_L - s_R & \text{turning right} \\ s_R - s_L & \text{turning left} \end{cases} \quad (4.6)$$

where B and C are constants such that $\min_s b^s = 0$ and $\max_s b^s = 1$. For the function f_θ we set $\theta = 1$. The default behavior of the robot is driving ahead along the line. The reflexes to turn to the right or left are triggered by the robots uncertainty about the current situation. This uncertainty can

be derived from the motor-map only and is naturally given by its entropy E , defined as

$$E = \log_2 \left(\sum o^{\mathbf{s}} \right) - \frac{1}{\sum o^{\mathbf{s}}} \sum o^{\mathbf{s}} \log_2(o^{\mathbf{s}}) \quad (4.7)$$

where the sums run over all \mathbf{s} . In a situation, where one single motor unit is active, the entropy becomes minimal, i.e. $E = 0$, which corresponds to minimal uncertainty. On the opposite, maximal uncertainty is reflected in the motor-map by a uniform activity distribution, which leads to maximal entropy. We hypothesize that entropy will be rather low while following lines and increase significantly at crossings, where several valid motor-actions are possible. Therefore, we define a dynamic threshold θ_E for the entropy, above which the default reflex “driving straight” will be replaced by one of the two turning reflexes, thus making the robot to turn to the side favored by the reflex. The threshold θ_E is given by

$$\theta_E(t) = \langle E \rangle_t + \delta \sqrt{\langle E^2 \rangle_t - \langle E \rangle_t^2} \quad (4.8)$$

where $\langle \cdot \rangle_t$ denotes the running average as defined in eq. 4.3 with $\tau = 1000$ time-steps. $\delta = 1.1$ is a constant. Note that the second term computes a running standard deviation of E .

4.3 Results

4.3.1 Line following

In the following we present experiments performed with the virtual robot and subsequently show that the results generalize to the real-world. Fig. 4.2b-d show the robot’s trajectories during 10^6 time-steps for three sensorimotor mappings acquired by the robot during 5’000, 40’000 and 2’500’000 time-steps of learning respectively. The mapping is randomly initialized. While the robot is poorly following the line after 5’000 time-steps of learning, its performance after 40’000 time-steps almost corresponds to the performance after 2’500’000 time-steps. Thus, these results qualitatively show that the learning mechanism for the acquisition of the sensorimotor mapping is both fast and stable. From a behavioral point of view, fig. 4.2

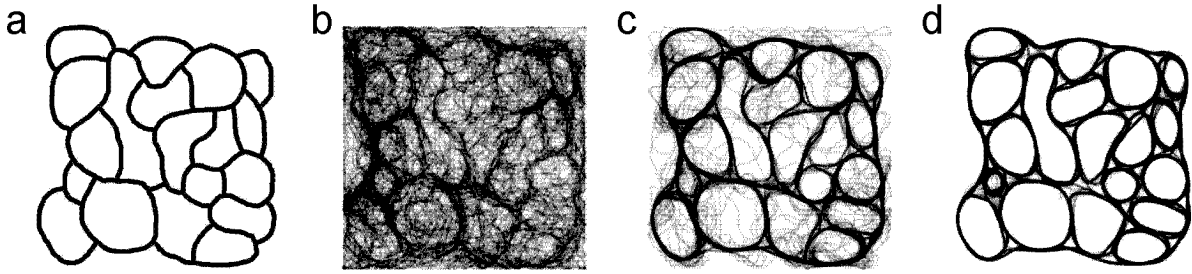


Figure 4.2: Virtual maze. (a) The maze of lines for the simulated robot consists of 36 crossings, each consisting of three lines. The average thickness of the lines is ≈ 0.5 body radii. (b-d) Trajectories of the robot recorded over 10^6 time-steps after the robot was learning to follow the line for 5'000, 40'000 and 2'500'000 time-steps.

illustrates the fact that the robot's movements become more controlled over time, i.e. the entropy in the robot's behavior reduces.

4.3.2 Motor-map entropy

We now investigate how this behavioral change is reflected in the internal states of the robot. The entropy of the motor-map as a function of time-steps the robot was learning is shown in fig. 4.3a. The average entropy decays over time. This reduction of the average entropy indicates that the initially broad activity in the motor-map in response to any given visual input stimulus gets sharpened and adjusted to the specified task. Thus the increased precision in line following is reflected in the reduced entropy in the motor-map.

4.3.3 Crossing detection

While the reduction of behavioral entropy leads to a stereotyped and reproducible behavior at a local level, a more flexible behavior is desirable at a global level. This flexibility, however, can only be achieved when the constraints given by the local behavior are relaxed, i.e. at crossings of lines. Indeed looking at the entropy of the motor-map we observe a persistently large variance (see fig. 4.3a). This reflects the property of the environment

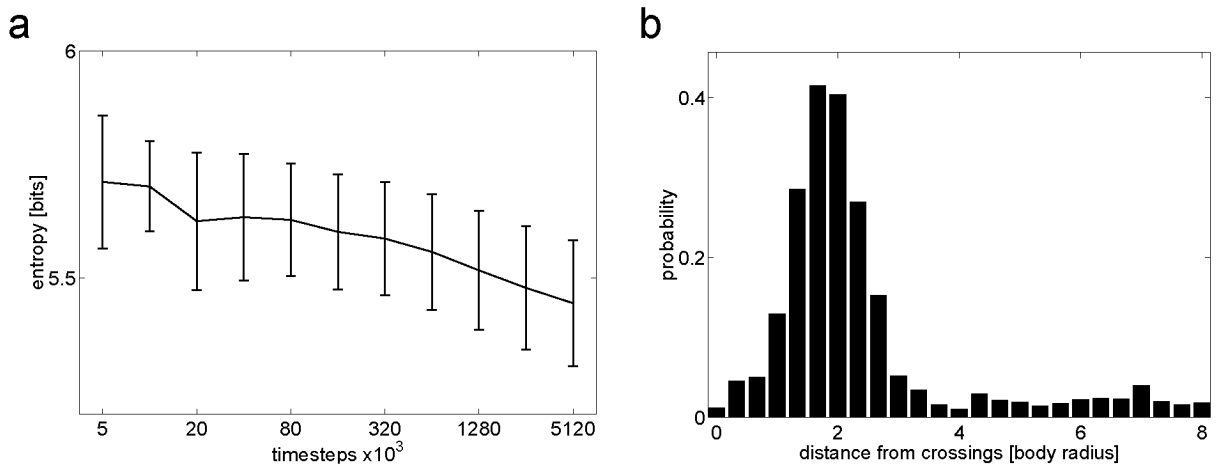


Figure 4.3: Entropy in the motor-map. **(a)** The average entropy decreases as a function of the time-steps the robot learns the sensorimotor mapping. The standard deviation (error bars) stays relatively constant ($\pm 10\%$). **(b)** The probability to trigger a crossing event as a function of the robot's distance from the crossing. A crossing is triggered as soon as the entropy in the motor-map exceeds a dynamic threshold which is determined by the running average and variance of the entropy (see *Methods*).

to not only consist of single lines but to also incorporate crossings. As a crucial ingredient to the proposed model, we use high entropy in the motor-map as a signature of a choice of multiple behaviors being available, i.e. a behavioral decision being required. Therefore, we use a simple criterion for the detection of crossings, namely the entropy of the motor-map exceeding a (dynamic) threshold (see *Methods*). Fig. 4.3b shows the probability that the robot detects a crossing based on this criterion as a function of the robots distance to the next crossing. The distribution peaks at a distance of approximately two body radii which corresponds to the distance at which the robot usually sees the crossing in the center of its visual field. Thus, the entropy in the motor-map is a reliable indicator of crossings.

4.3.4 Decision making

In the following we use this signal to trigger random reflexes in order to bias the robot's movements towards a particular direction. Therefore, the default activity ramp imposed on the motor-map for driving forward is replaced by a ramp favoring either right or left movements. Both directions have the same probability to be chosen. We investigate these principles in the example of the particular crossing shown in fig. 4.4 where the robot approaches from the bottom left corner. After 10'000 time-steps of learning, the crossing is still a likely source of errors where the robot loses the line (fig. 4.4a). In addition, the robot follows the right branch more often although the decision making mechanism chooses both directions with equal probability (see fig. 4.4d). After further learning (40'000 time-steps), the errors are reduced, but there is still an imbalance in the directions the robot takes at the crossing (fig. 4.4b). This imbalance is determined by the particular morphology of the crossing and the direction from which the robot approaches. More importantly, however, this shows that at this stage the robot's sensorimotor mapping is not yet accurate enough for the given task. Thus, either the robot does not detect the crossing or it fails to execute a decision appropriately. It is not until a further refinement of the sensorimotor mapping, that the entropy in the motor-map becomes a reliable indicator of crossings and the motor-map itself allows any branch to be followed. The previous imbalance is evened out by taking and executing random decisions rather than following the more "convenient" branch (see fig. 4.4c,d). Thus, we exploit the events of high entropy to trigger the

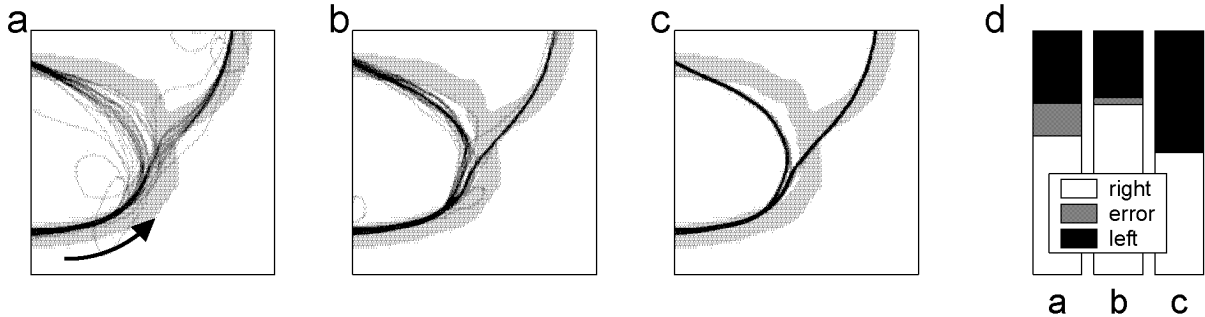


Figure 4.4: Sample crossing. **(a-c)** Trajectories of the robot approaching from the lower-left branch (see arrow) after 10'000, 40'000 and 2'500'000 time-steps of learning respectively. **(d)** Fractions of left turns, right turns and errors at the crossing for the situations shown in (a-c). $N = 37, 33, 32$ respectively.

transitions from reproducible and stereotyped behavior at a local level to flexible and explorative behavior at a more global level.

The entropy in the motor-map reliably reflects the complexity of the behavioral context. Crossings elicit broader activity within the motor-map, because both branches, respectively their sensorimotor mappings, are represented simultaneously. This effect has two implications: (1) the entropy of the motor-map increases which indicates the increased complexity of the behavioral context (2) coarse interaction within the motor-map by applying activity ramps can be used to impose general decisions such as taking the right or left branch without specifying in detail what needs to be done in order to do so.

We now show that the previous findings, based on the single crossing shown in fig. 4.4, hold for the whole maze. In total, the maze contains 36 crossings which can be approached from three different branches each. Thus, the maze consists of 108 oriented branches, which should be visited equally often given that the robot makes unbiased decisions at each crossing. Fig. 4.6a shows the rates at which the robot visits the different oriented branches. The control experiment, in which the decision making mechanism was not used, shows a large imbalance, e.g. some oriented branches are not visited at all, while others are visited more than three times more often than expected for a uniform maze coverage. Using the decision mak-

ing mechanism reduces this imbalance greatly and in particular there are no more unvisited branches. We captured the imbalance of the different distributions by calculating the standard deviation over the rates of visiting the oriented branches. In fig. 4.6b we show this measure of inhomogeneity with and without decision making mechanism. The robot’s coverage of the maze is more homogeneous using the decision making mechanism. Thus, the above result generalizes to a wide variety of crossing morphologies.

While the above investigation has shown that the decision making mechanism has a great impact on the global behavior of the robot, it is unclear what possible other effects there are. For example, investigating the speed profile of the robot running around in the maze reveals interesting characteristics. First, with decision making turned on, the shape of the speed distribution changes for different durations of the learning periods (fig. 4.5a-c). While the distribution appears to be rather symmetric around a medium speed for 10’000 and 40’000 time-steps of learning, it becomes very skewed towards higher speeds for 2’500’000 time-steps of learning. Thus, longer periods of learning lead to higher average speeds. Comparing, however, the situation where the decision making mechanism is turned on (fig. 4.5c) to where it is turned off (fig. 4.5d) shows a very prominent difference. While both distributions peak at approximately the same speed, the distribution with decision making is less skewed indicating that lower speeds are driven more often. This increase in low speeds is due to the fact that decision making may lead to situations where the robot has to take very sharp turns which are usually driven at lower speeds. Without decision making, however, the robot only drives along the branches which are most “convenient”, i.e. the branches which require a minimal amount of turning and therefore can be followed at larger speeds. Thus, the decision making mechanisms does not only influence the robots behavior at a global level, but also has an effect on the robot’s local movement statistics.

The results discussed so far originate from the simulated virtual robot. This setup guarantees repeatability over trials and therefore allows for a systematic evaluation of the proposed model. Only experiments with a real robot, however, allow the exploration of the robustness and the generalizability of a model [55]. Therefore, the above experiments are also performed in the real-world using the micro-robot Khepera. After one hour of learning ($\approx 90’000$ time-steps), the precision of line-tracking reaches a high level. Analyzing maze coverage reveals that the homogeneity

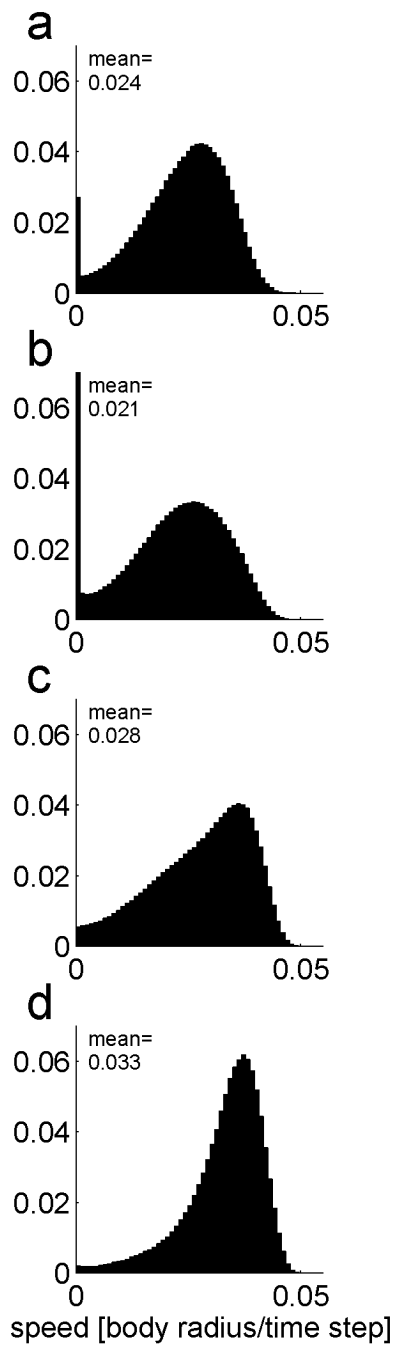


Figure 4.5: **(a-c)** Distribution of the robot's driving speed after 10'000, 40'000 and 2'500'000 time-steps of learning respectively. **(d)** Speed distribution without decision making.

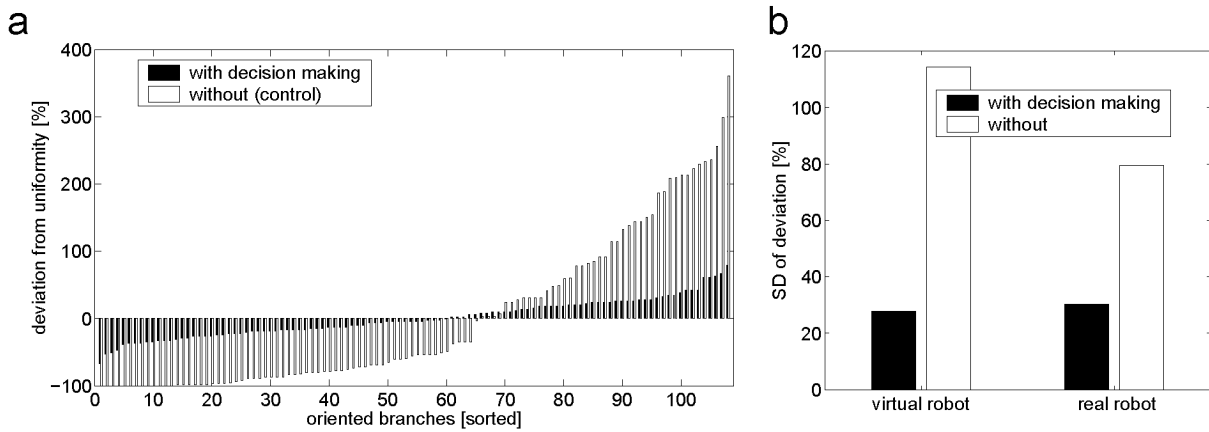


Figure 4.6: Maze coverage and inhomogeneity. **(a)** The deviation from uniform maze coverage for a robot using the decision making mechanism (black bars) and a control, for which no random decisions are imposed (white bars). For both distributions, the oriented branches were sorted according to their deviation from uniformity. The sorting order is not necessarily identical for the two distributions. **(b)** Inhomogeneity for the virtual and the real-world robot with and without decision making (black and white bars respectively). Inhomogeneity is defined as the standard deviation of the rates at which the different oriented branches are visited in percent of the uniform rate. Please note that this measure is independent of the total number of crossings.

is greatly increased by the decision making mechanism (fig. 4.6b). Thus, the performance of the real-world robot matches the simulated system on the local as well as the global behavioral level.

4.4 Discussion

We used a neural model of sensorimotor integration to study to what extent the motor system itself can contribute to the interpretation of behavioral situations. The proposed neural network controls the behaving robot which learns to perform in a navigation task, i.e. line following. We show that the population response of the motor system provides a substrate for the categorization of behavioral situations. Based on this categorization, higher level decision making is engaged in order to resolve behavioral conflicts. As a result this self-contained model achieves optimal balance between local and global requirements of the behavioral task.

It is unclear how the concept put forward in this study of detecting behavioral salient events through the motor system, generalizes to behavioral paradigms other than line following which is a strongly constrained type of navigation; it can be accomplished using exclusively local information in space and time, i.e. information that can be acquired from local sensors and processed in a purely feed-forward fashion (no memory). Apart from this however, there is no a priori assumption specific to line following. Thus, we claim that the above concept holds for other behavioral paradigms as well, as long as the underlying sensorimotor mapping is learned by the agent autonomously.

As a direct consequence of an agent learning a behavioral task, its behavioral diversity is reduced. Thus, in the case of line following, the robots movements along the line become more and more accurate and reproducible in order to fulfill the goal of keeping the gaze centered on the line. While this might be very desirable from a purely local point of view, the agent may be confronted with different if not conflicting behavioral goals at more global levels. In order to optimally explore a maze of lines, an agent rather needs to diversify its behavior selectively in those situations where the constraints imposed by the line following are relaxed, i.e. at crossings. It is therefore crucial to be able to identify these situations, in order to maintain an optimal balance between different requirements

of a behavioral task. In this study we have shown, that the population response of the motor system provides the relevant information in order to detect such situations. Moreover, incorporating a simple decision making mechanism, the robot has shown to optimally perform both at a local level, i.e. accurate line following, as well as at a global level, i.e. even coverage of the maze.

The linkage between sensory input and behavior involving interpretation and behavioral selection is usually referred to as a decision process. Currently, no dedicated brain structure is known, which contains an abstract representation of interpretations or decisions, without being related to a motor system or dependent on continuous sensory stimulation [46]. Furthermore, the cortical areas along the path from primary sensory to primary motor areas form a reciprocally coupled network where information flows in either direction. Here, we hypothesize that the decision process is an emergent property of this bidirectional interaction. One important ingredient to a decision process is the detection of situations at which decisions may be required. A theoretical study suggests that sensory neurons can encode the variables required to compute a decision, but they neither carry out this computation nor represent its outcome [77]. In the case of detecting behaviorally relevant situations, the motor system might be a more adequate candidate. Given that the sensorimotor mapping is learned with an appropriate learning rule with no a priori assumptions about the requirements for a particular task, behaviorally ambiguous situations are naturally translated into “ambiguous activation” of the motor-map. Put another way, in order to detect behaviorally ambiguous situations one needs to know what the different behavioral options are. Since the set of possible behavioral choices at each moment in time is strongly constrained by the characteristics of the agent’s motor system, control structures closer to the motor end of the sensorimotor mapping are better qualified for the detection of behaviorally ambiguous situations. In addition, the dimensionality and variability of a perceived environment usually surmounts the behavioral diversity of an animal, such that interpretations of behavioral situations might be more efficiently guided by the low dimensional but task-specific representation of the world at the motor end. Therefore, the motor cortex can be considered as part of a high-level perceptual system [70].

Chapter 5

Temporal population codes and the formation of place fields

In the last chapter of the main body of this thesis, we would like to discuss how the properties of the enhanced temporal population code presented in chapter 3 can be exploited by higher order systems to form allocentric representations of a physical environment based on purely visual egocentric sensory information. The property required for this non-trivial transformation is bounded invariance.

5.1 Introduction

Cells in the hippocampus of the rat fire selectively depending on the rat's position within the environment irrespective of its orientation. The ensemble of locations where such a cell fires – the *place field* – is determined by a combination of different environmental and internal cues [60]. The question arises, however, what kind of computations are performed in order to map egocentric sensory information about various cues to an allocentric representation of space. O'Keefe and Burgess proposed that a place field is formed by the summation of Gaussian tuning curves, each oriented perpendicular to a wall of the environment and peaked at a fixed distance from it [18, 58, 59]. While this proposal tries to explain the actual transformation from one coordinate system to another, it does not account for the

problem on how to form egocentric representations of space which could be appropriate for the proposed computation. Thus, it is yet unclear, how the information about a rat's distance to different walls becomes available, and in particular how this proposal would generalize to other environments where more advanced visual skills, such as cue identification, are required.

Considering the role of the visual sensory modality and its role in providing such an appropriate egocentric representation leads to the notion of bounded invariance. For an agent moving in an environment, visual percepts of objects may undergo a combination of transforms comprising zooming and shearing (rotation in depth). Thus in order to distinguish two views of the same visual object seen from different positions within the environment one needs not to be invariant with respect to zooming and shearing. In contrast, non-invariance also means reduced or no generalizability, i.e. small variations between two visual percepts of the same object prevents them to be identified. A path between the two extremes above is bounded invariance which provides both, generalization as well as specificity. In terms of place coding, this means that the position from which a particular visual object is perceived can be identified simply by comparing to visual percepts of the same object from other locations. Actually, the property of bounded invariance is tightly coupled to the internal representation of the visual percepts being topology preserving, i.e. views of the same object from locations nearby result in similar representations while their representations become very dissimilar if the locations are far apart.

In this chapter, we show that model for visual pattern recognition proposed in the first two chapters of this thesis supports the notion of bounded invariance. Compared to other approaches towards invariant pattern recognition [26,45], the temporal population code is not a pattern classifier in the traditional sense, but rather a transformation which is topology preserving, i.e. similar visual stimuli result in similar temporal population codes. Thus, the visual similarity between different stimuli is directly accessible from the temporal population code without the need for learning. This is in contrast to other approaches towards invariant pattern recognition which require extensive supervised training sessions for all possible combinations of stimuli in order to make any information about their similarity accessible.

We investigate to which extent the properties of the temporal popula-

tion code can be exploited for place field formation using a virtual robot which explores its environment with a camera mounted on its top. We show that the similarity between two different views of the same stimulus directly translates to the topology of the temporal population coding space. This guarantees for favorable generalization properties across different views of the same stimuli which makes extensive scanning of the stimulus space unnecessary. We will show that a robot can form place-fields from a series of temporal population code snapshots of the salient visual cues in its environment. The response of such a place cell degrades smoothly and monotonically for increasing distances from the location where it was established. This is due to the bounded invariance properties of the temporal population code and allows for a stable and accurate position reconstruction given the responses of a small number of place cells.

5.2 Methods

Experiments were performed using a simulated version of the real-world robot *Khepera* (K-team, Lausanne, Switzerland) programmed in C++ using OpenGL (sec. A.3.3). The robot has a circular body with two wheels attached to its side each controlled by an individual motor. The visual input is provided by a camera with a viewing angle of 60° mounted on top of the robot. The neural networks are simulated on a Linux computer using a neural network simulator programmed in C++ (see Appendix A).

The environment consists of a square arena with a gray floor surrounded by white walls (fig. 5.1, left). In the following, all lengths will be given in units of the side lengths of the square environment. Four different black patterns are drawn on the walls approximately at the height of the robot's camera. These patterns, a square, a triangle, a Z and a X constitute the visual cues from which the robot forms a representation of space.

5.2.1 The temporal population code

The visual information captured by the camera is processed by the cortical type network proposed in chapter 3. The input image first passes through an edge-detection stage by convolving it with a difference of Gaussian (DOG) kernel. The resulting contour represents the activity in the

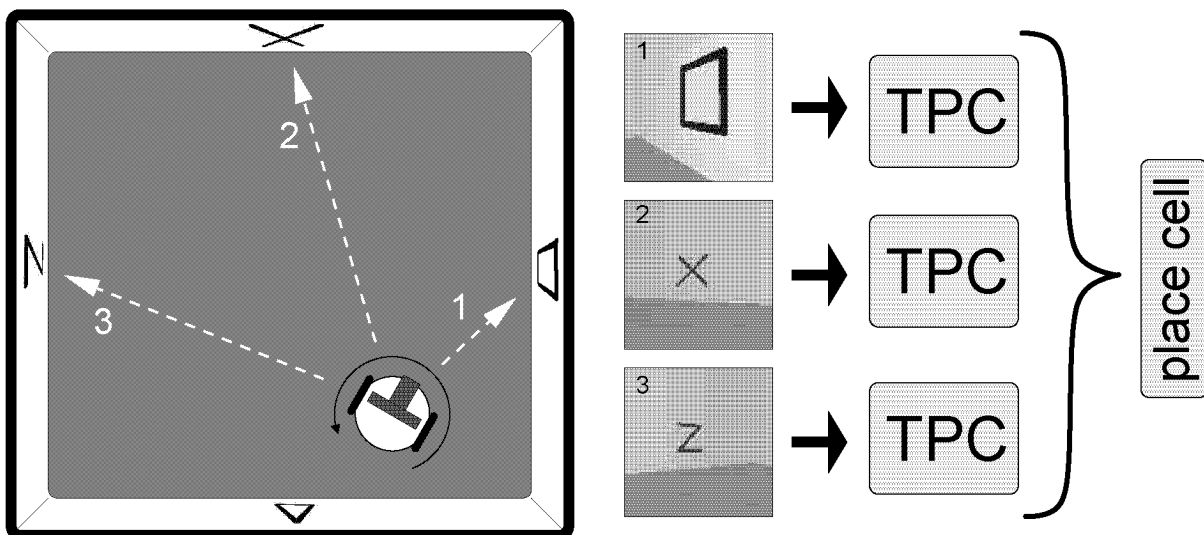


Figure 5.1: Place cells from multiple snapshots. The robot is placed in a virtual square environment with four black patterns on the white walls, i.e. a square, a triangle, a Z and a X. The robot scans the environment for salient stimuli by rotating on place. A simple saliency detector triggers the acquisition of visual snapshots which are subsequently transformed into temporal population codes (TPC). A place cell is defined through its associated temporal population code templates.

LGN, which serves as the input to the cortical network. This network consists of cortical columns which are selective for one of four orientations (0° , 45° , 90° and 135°) and one of three spatial frequencies (high, medium and low). The lateral connectivity between the columns is exclusively excitatory and depends on their respective selectivities. While columns of different spatial frequency selectivities do not interact, the type of connectivity between columns of the same spatial frequency selectivity depends on the orientation selectivity. For short distances columns connect to any other column irrespective of their orientation selectivity or their relative position. Long-range connections are permitted exclusively between columns with the same orientation selectivity. Furthermore, these columns must be arranged along the common preferred orientation in order to become connected. For further details about the network, please see section 3.2.1 of this thesis.

The output of the network is given by twelve 100-dimensional vectors $\mathbf{A}_{\psi,\nu}$ with $\psi \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ and $\nu \in \{\text{high, medium, low}\}$ which reflect the average population activity recorded over 100 time-steps for each type of cortical column. In the following we concatenate the four corresponding vectors with the same spatial frequency selectivity leaving us with three 400-dimensional vectors \mathbf{A}_ν . This set of vectors form the temporal population code within which visual information of a snapshot will be represented.

The similarity $S(s_1, s_2)$ between two snapshots s_1 and s_2 is defined as the average correlation between the corresponding vectors, i.e.

$$S(s_1, s_2) = \left\langle Z\left(\rho(\mathbf{A}_\nu^{s_1}, \mathbf{A}_\nu^{s_2})\right) \right\rangle_{\forall \nu} \quad (5.1)$$

$$\text{where } \rho(\mathbf{A}_\nu^{s_1}, \mathbf{A}_\nu^{s_2}) = \frac{\text{cov}(\mathbf{A}_\nu^{s_1}, \mathbf{A}_\nu^{s_2})}{\sqrt{\text{var}(\mathbf{A}_\nu^{s_1})\text{var}(\mathbf{A}_\nu^{s_2})}$$

where Z is the Fisher Z-Transform introduced in section 2.2.2.

5.2.2 Place cells from multiple snapshots

In this study, the response properties of a place cell are given by the similarity between incoming snapshots of the environment and template snapshots

associated to the place cell when it was constructed. Thus, for both, the acquisition of place cells as well as their exploitation, the system needs to be provided with informative snapshots of its environment. For this purpose, the robot is equipped with a simple visual saliency detector based on a heuristic saliency trace $s(t)$ which captures the relative contrast within the central visual field. Thus, this trace is determined by the activity $a(\mathbf{y}, t)$ of the cells in the LGN, where $\mathbf{y} \in [-1, +1]^2$ defines the position of the cell within LGN (assuming the LGN cells lie on the nodes of a regular lattice covering the $[-1, +1]^2$ -plane):

$$s(t) = \frac{\sum e^{-y^2} a(\mathbf{y}, t)^2}{\sum a(\mathbf{y}, t)^2}$$

where both sums run over all \mathbf{y} . Thus, $s(t)$ becomes large for relatively high contrast in the central visual field as compared to the periphery. At each point in time where $s(t) > \theta_{saliency}$, a new snapshot is acquired with a probability of 0.1. A minimal delay between subsequent snapshots is imposed in order to prevent several snapshots to be taken from the same gaze. A place-cell k is defined by several snapshots acquired at the same location but for different orientations. These snapshots will be called templates t_i^k with $i = 1 \dots n$, where n is the number of templates stored per place cell.

Whenever the robot tries to localize itself, it scans the environment by rotating in place and taking snapshots of visually salient scenes (fig. 5.1). The similarity S between each incoming snapshot s_j with $j = 1 \dots m$ and every template t_i^k is determined using eq. 5.1. The activation a_k of place-cell k for a series of m snapshots s_j is then given by a sigmoidal function

$$a_k(i_k) = \left(1 + \exp\left(-\beta(i_k - \theta)\right) \right)^{-1} \quad \text{where} \quad i_k = \left\langle \max_i \left(S(t_i^k, s_j) \right) \right\rangle_j. \quad (5.2)$$

i_k represents the input to the place-cell which is computed by determining the maximal similarity of each snapshot to any template of the place cell and subsequent averaging, i.e. $\langle \cdot \rangle_j$ corresponds to the average over all snapshots j .

5.2.3 Position reconstruction

Given the response of individual place cells, the objective in the following is to investigate the accuracy with which the robot's position is encoded by a population of place cells. There are many different approaches to the problem of position reconstruction or decoding from place cell activity [104]. Methods like population vectors [29], template matching [97] or the optimal linear estimator [74] are special cases of the more general basis function framework. A basis function method uses a linear combination of basis functions $\phi_k(\mathbf{x})$ with the coefficients proportional to the activity of the place cells a_k , i.e.

$$\sum_k a_k \phi_k(\mathbf{x}).$$

Here we use a direct basis approach, i.e. the basis function $\phi_k(\mathbf{x})$ directly corresponds to the average activation a_k of place cell k at position \mathbf{x} within the environment. The reconstructed position $\hat{\mathbf{x}}$ is then given by

$$\hat{\mathbf{x}} = \operatorname{argmax}_{\mathbf{x}} \sum_k a_k \phi_k(\mathbf{x})$$

The reconstruction error is given by the distance between the reconstructed and true position averaged over all positions within the environment.

5.2.4 Place field shape and size

In order to investigate the shape of a place field $\phi(\mathbf{x})$, and in particular to determine its degree of asymmetry and its size, we computed the two-dimensional normalized inertial tensor \mathbf{I} given by

$$I_{ij} = \frac{\sum_{\mathbf{r}} \phi(\mathbf{r}) (\delta_{ij} \mathbf{r}^2 - r_i r_j)}{\sum_{\mathbf{r}} \phi(\mathbf{r})}$$

with $\mathbf{r} = \{r_1, r_2\} = \mathbf{x} - \hat{\mathbf{x}}$ where $\hat{\mathbf{x}} = \sum \mathbf{x} \phi(\mathbf{x}) / \sum \phi(\mathbf{x})$ corresponds to the ‘‘center of gravity’’. δ_{ij} is the Kronecker delta. \mathbf{I} is symmetric and can therefore be diagonalized, i.e. $\mathbf{I} = \mathbf{V}^T \mathbf{D} \mathbf{V}$, such that \mathbf{V} is an orthonormal transformation matrix and $D_{ii} > 0$ for $i = 1, 2$. A measure of the half-width of the place field along its two principal axes is then $d_i = \sqrt{2D_{ii}}$

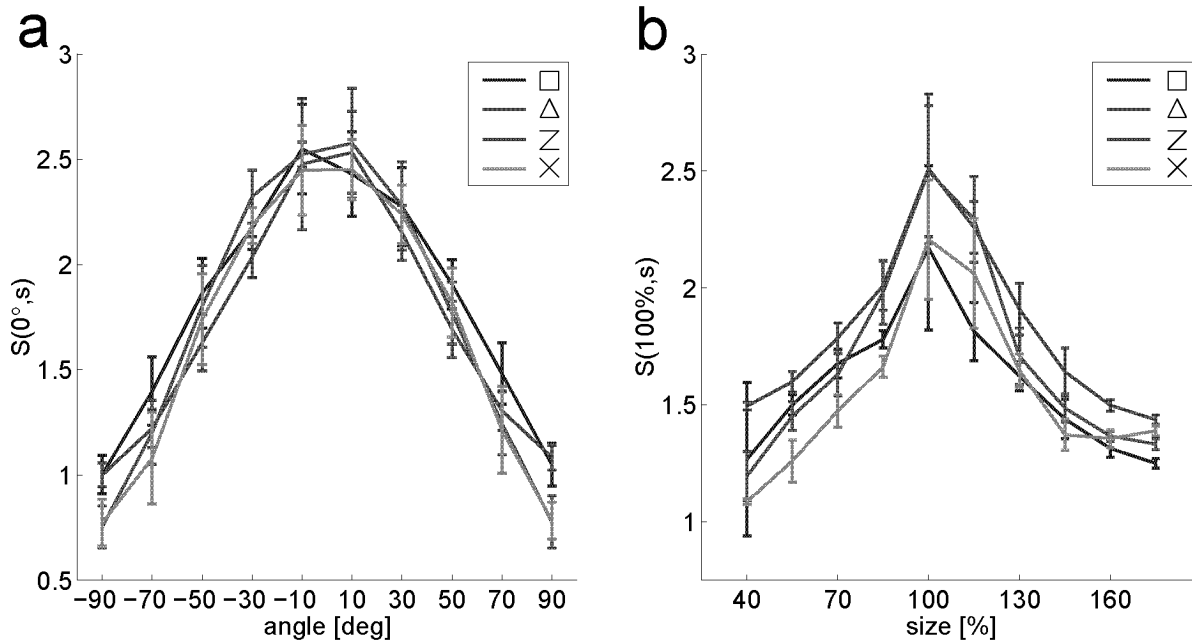


Figure 5.2: Tolerance with respect to viewing angle and distance. The similarity between pairs of the same stimuli (square, triangle, Z or X) under different viewing angles (a) and distances (b). Note that in (b), similarity is plotted as a function of relative stimulus size which is inversely proportional to viewing distance.

such that a measure of asymmetry is given by

$$0 \leq \left| \frac{d_1 - d_2}{d_1 + d_2} \right| \leq 1$$

This measure becomes zero for symmetric place fields while approaching one for asymmetric ones. In addition, we can estimate the size of the place field by approximating its shape by an ellipse, i.e. $\pi d_1 d_2$.

5.3 Results

5.3.1 Bounded invariance

Initially, we would like to investigate the topological properties of the temporal population coding space. In particular we are interested in how

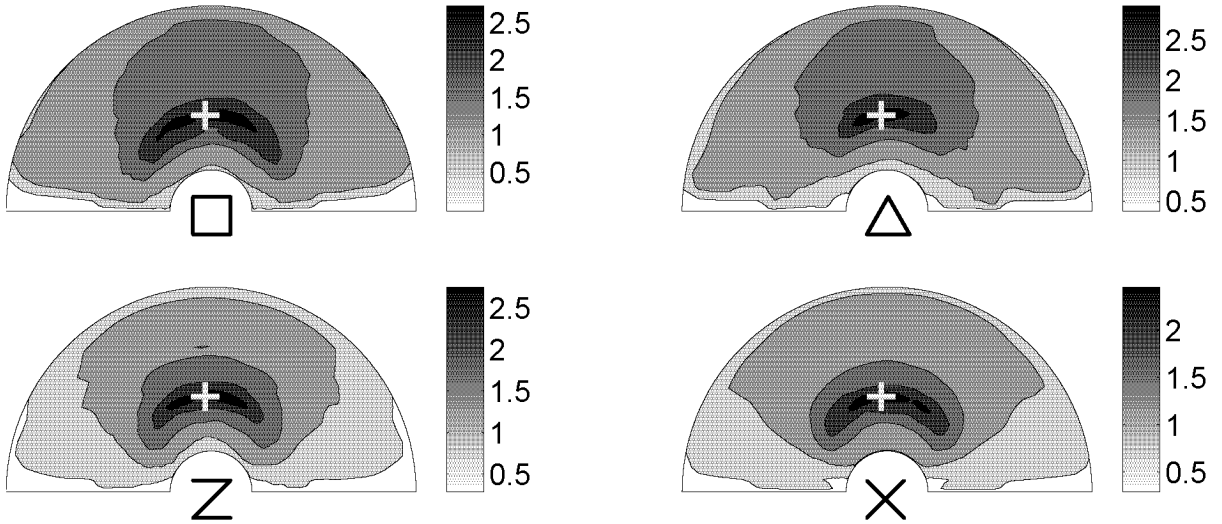


Figure 5.3: Similarity surface for the four different cues. Similarity between a reference snapshot of the different cues taken at the position marked by the white cross and all the other positions surrounding the reference location.

environmental neighborhood relationships are translated and represented in the similarity between temporal population code snapshots. This is illustrated in fig. 5.2 where the similarity S (eq. 5.1) between a pair of snapshots of the same stimuli under different viewing angles (fig. 5.2a) and distances (fig. 5.2b) is plotted. Stimuli which are visually similar have similar temporal population codes. The bell-shaped tuning of the similarity measure shows the bounded invariance of the temporal population code with respect to the two geometric transforms, shearing (rotation in 3D) and scaling.

In the experimental setup investigated subsequently, the robot scans for cues on the walls. Depending on the robot's position within the environment, the different cues undergo a geometric transformation which is a combination of scaling and shearing. Fig. 5.3 shows the similarity to a reference snapshot taken at the location of the white cross for the four different cues. Although the precise shape of the similarity surface differs for the different stimuli, the similarity decreases smoothly and monotonically for increasing distances to the reference point for all stimuli.

The similarity surface for different locations of the reference point is

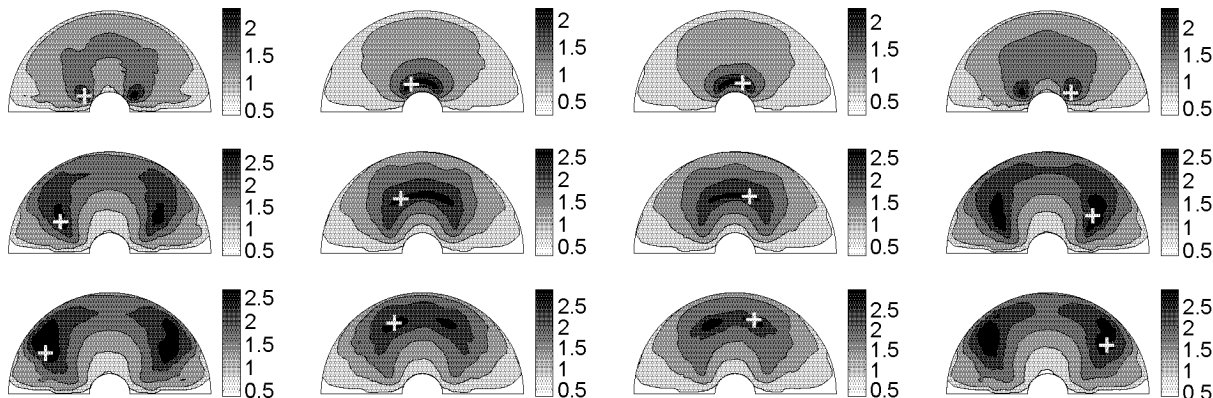


Figure 5.4: Similarity surface of Z cue for different reference points. The distance/angle of the reference point to the cue is kept constant along the rows/columns respectively.

shown in fig. 5.4 for the Z cue. Although the Z cue has no vertical mirror symmetry, the similarity surfaces are nearly symmetric with respect to the vertical center line. Thus, using a single cue, localization is only possible modulo a mirror along the vertical center. The implications of this will be discussed later. Concerning different distances of the reference point to the stimulus, fig. 5.4 (along the columns) shows that the specificity of the similarity measure is large for small distances while the tuning becomes broader for large distances. This is a natural consequence of the perspective projection which implies that the size of a stimulus is inversely proportional to the viewing distance. Thus, at large viewing distances the size of the stimulus changes little for small movements around the reference point. At small distances, however, small movements can yield dramatic changes in stimulus size such that the similarity measure becomes very specific.

5.3.2 Place cells from multiple snapshots

In the following we define a place-cell as a unit whose response properties are determined by eq. 5.2 based on four associated snapshots/templates taken at the same location within the environment. In the following, the location at which the templates for a place-cell are recorded will be called the “associated” location. The templates for each place-cell are chosen by the stochastic saliency detector and therefore there is no explicit control

over the actual snapshots defining a place-cell, i.e. some place cells are defined based on two or more templates of the same cue. Furthermore, the stochastic nature of the saliency detector does not allow for any control over the precise position of the stimulus within the visual field. This is, where the intrinsic translation invariance of the temporal population code plays an important role, i.e. the precise position of the stimulus within the visual field at the time of the snapshot has no effect on the resulting encoding as long as the whole stimulus is visible.

Fig. 5.5 shows examples of the receptive fields (subsequently also called place fields) of such place cells acquired at the nodes of a regular 5×5 lattice within the environment. Most of the place fields have a Gaussian-like tuning which is compatible with single cell recordings from pyramidal cells in CA3 and CA1 [60], i.e. the place cells maximally respond close to their associated positions and degrade smoothly and monotonically for increasing distances. Some place cells have multiple subfields in that they respond to different locations in the environment with a similar amplitude. In accordance with experimental data [59], there appears to be a tendency for place fields in the center to be more symmetric than those towards the borders of the environment. A more detailed investigation regarding place field shape will follow.

5.3.3 Position reconstruction

Subsequently, we determine the accuracy up to which the robot can be localized within the environment as a function of the number of place cells and the number of snapshots taken per location. Therefore we use the direct basis approach for position reconstruction as described in section 5.2.3. As basis functions we take the normalized response profiles of place cells constructed from four templates at the nodes of a regular lattice of varying resolution covering the environment. Fig. 5.6 shows the reconstruction error averaged over the environment as a function of the number of place cells as well as the number of snapshots taken at each location. The reconstruction error decreases monotonically both for an increasing number of place cells as well as an increasing number of snapshots. An asymptotic reconstruction error is approached very fast, i.e. for more than 25 place cells and more than two snapshots per location. Thus, for a behaving organism exploring an unknown environment, this implies that a relatively

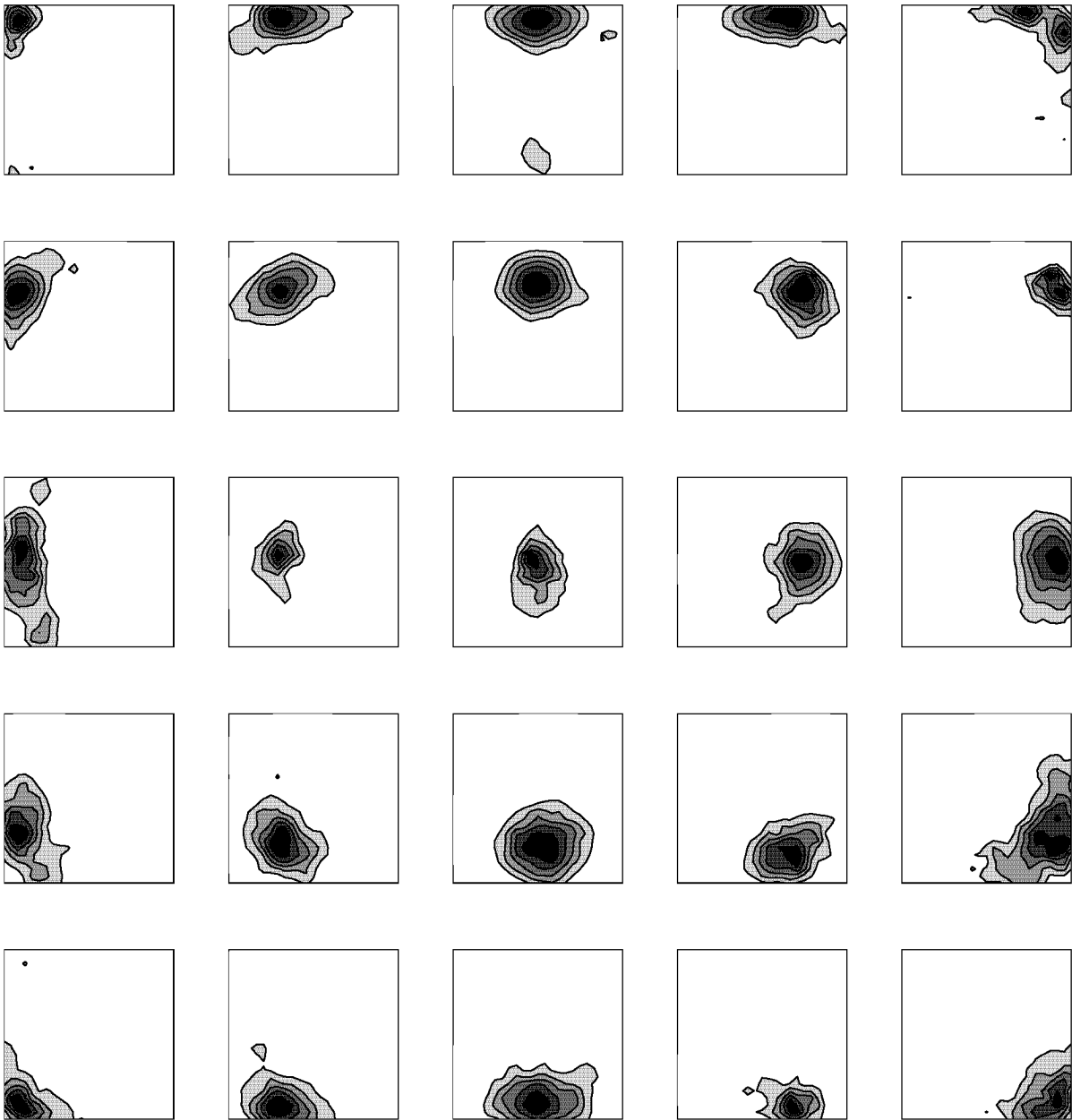


Figure 5.5: Place fields of 5×5 place cells. The average response of 5×5 different place cells for all the positions of the robot within the environment. Darker regions correspond to stronger responses. The relative location of each place field within the figure corresponds to the associated location of the place-cell within the environment. All place fields are scaled to a common maximum response.

sparse exploration strategy suffices to create a complete representation of the new environment.

We now would like to analyze the dependency of the reconstruction error on the number of snapshots in greater detail. Above we have seen that localization with a single snapshot is only possible modulo a mirror along the axis where the cue is located. The systematic reconstruction error introduced by this short-coming can be determined analytically and is ≈ 0.13 in units of the side-length of the square environment. For an increasing number of snapshots, the probability that all snapshots are from the same pair of opposite cues, decreases exponentially fast and we therefore also expect the systematic error to vanish. Considering 100 place cells, the difference in reconstruction error between 1 and 10 snapshots amounts to 0.147 ± 0.008 (mean \pm SD) which is close to the predicted systematic error due to the effect discussed above. Thus, an increasing number of snapshots primarily helps to resolve ambiguities due to the symmetry properties of the temporal population code.

5.3.4 Place field shape

The place field shape is analyzed by computing its two-dimensional normalized inertial tensor from which the place field asymmetry and size can be estimated as described in section 5.2.4. Fig. 5.7 shows scatter-plots of both, place field asymmetry and size versus the distance of the place field's associated location from the center of the square environment. There is a tendency that off-center place cells have more asymmetric place fields than cells closer to the center ($r=0.32$) which is in accordance with experimental results [59]. Regarding place field size, there is no direct relation to the associated position of place field ($r=0.08$) apart from the fact that the variance is maximal for intermediate distances from the center. It must be noted, however, that the size of the place field critically depends on the choice of the threshold θ in eq. 5.2. Indeed different relations between place field size and location can be achieved by assuming non homogeneous thresholds, which for example might be determined for each place cell individually based on its range of inputs. The measure for place field asymmetry, in contrast, has shown to be more stable in this respect (data not shown).

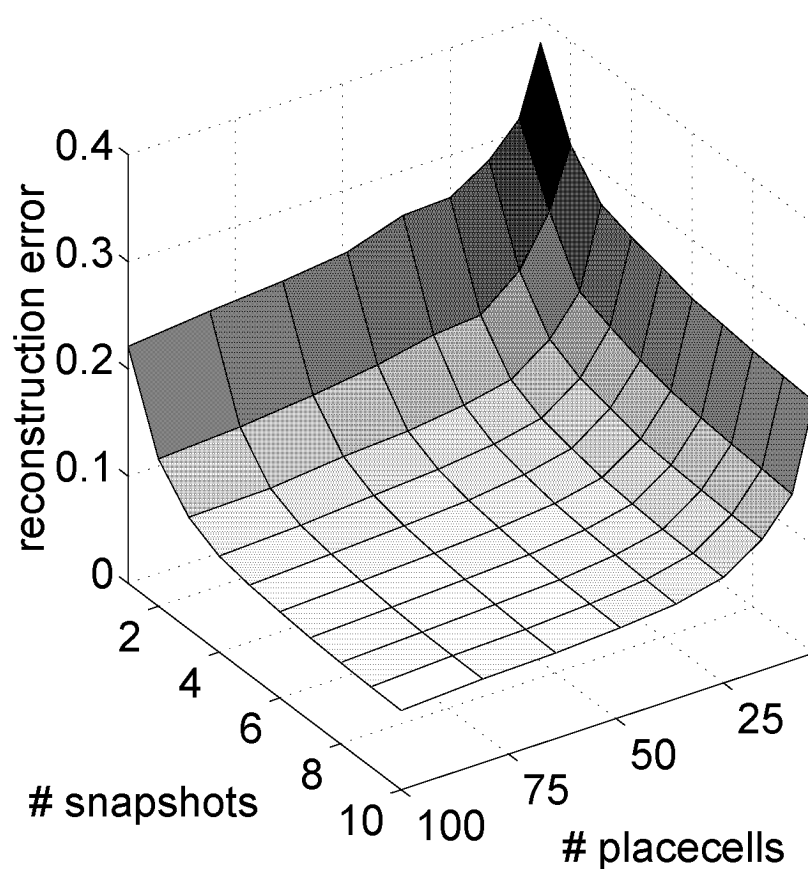


Figure 5.6: Position reconstruction error. The average error in position reconstruction as a function of the number of snapshots taken at each position and the number of place cells considered. The place cells used for reconstruction are located on the nodes of a regular lattice which spans the environment evenly.

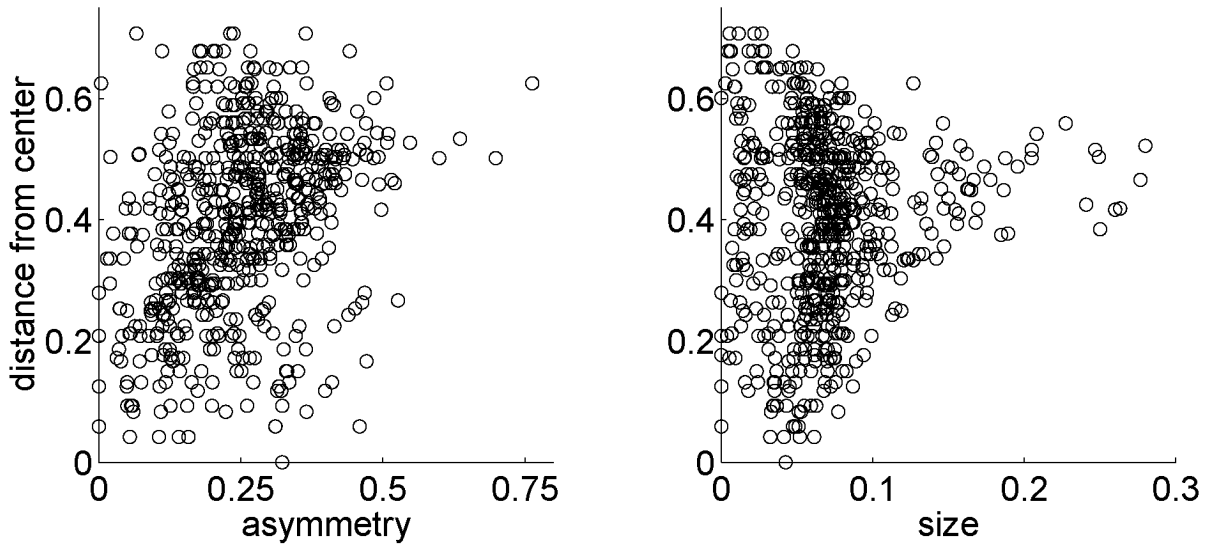


Figure 5.7: Place field asymmetry and size. Scatter plots of the place field asymmetry/size versus the distance of the place fields associated location to the center of the environment. The correlation coefficients are $r=0.32/0.08$ respectively.

5.4 Discussion

We have shown that the bounded invariance properties of visual stimuli encoded in a temporal population code are well suited for the formation of place fields. More specifically, the topology preservation of similarity amongst different viewing angles and distances allows a direct translation of the visual similarity between two views to their relative location within an environment. Due to the relatively broad tuning of the bounded invariances only a small number of place cells are required for self-localization, i.e. position reconstruction accuracy saturates for more than 25 place cells. Regarding the shape of the receptive fields of these place cells, only weak correlations between place field asymmetry/size and the distance of the place field to the center of the environment have been found.

As opposed to the present approach, experimental results suggest that place field formation in the hippocampus relies on multiple sensory modalities and not only vision. Although it was shown that vision may play an important role [40], proprioceptive stimuli, for example, can become important in situations where either visual information is not available such

as in the dark or in the presence of visual singularities, where two different locations elicit the same visual sensation [8]. A type of information which is very much related to proprioceptive stimuli, is the causal structure of behavior which imposes continuous movement in both space and time, i.e. the information about the last location can be of great importance for estimating the current location [64]. Indeed, a recent study has shown that position reconstruction error greatly reduces, if this additional constraint is taken into account [104]. In the present approach we analyzed the properties of place cells in the absence of a behavioral paradigm. Thus, it is not meaningful to integrate information over different locations. We expect, however, that for a continuously behaving robot this type of information would be particularly useful to resolve the ambiguities introduced by the mirror invariance in the case of a single visual snapshot.

As opposed to the large field of view of rats ($\approx 320^\circ$ [36]) the robot used in this study has a very restricted field of view of 60° . This has direct implications on the robots behavior. While a rat has probably continuous visual stimulation, the robot has to scan for visual stimuli. The advantage of only considering a 60° field of view is, however, that the amount of information contributed by single oriented views can be investigated. We have shown, that a single view allows for localization modulo a mirror along the orientation of the corresponding stimulus. This ambiguity can be resolved taking additional snapshots into account. In this context, maximal additional information can be gained if a new snapshot is taken along a direction orthogonal to the first snapshot. In addition, from a behavioral point of view, it would also be more efficient to compare snapshots of stimuli on neighboring rather than opposite walls, since the former would require smaller turn angles than the latter. Although, these considerations do not directly translate to the situation of a rat, they might still be relevant, since, as mentioned above, the rat has not complete panoramic vision, such that rotational movements are still required for two opposite snapshots, while no movement would be necessary at all in the case of two neighboring stimuli.

In the present study, the acquisition of place cells was supervised, in that their associated locations are assumed to correspond to the nodes of a regular lattice covering the environment. While this allows for a controlled statistical analysis of the place cell properties, it is not very likely that an autonomously behaving agent can acquire place cells in such a regular

fashion. It is more likely, that place cells have to be acquired incrementally based on purely local information. Information about the number of place cells responding or the maximal response of any place cell for a particular location is locally available to the agent, and can therefore be used to selectively trigger the acquisition of new place cells. In general, the representation will most likely also reflect further behavioral requirements in that important locations where decisions need to be taken, will be represented by a high density of place cells. Locations leaving little or no behavioral options will have accordingly fewer associated place cells. A possible solution to the problem of how an agent can bootstrap the notion of behavioral relevance has been proposed recently [99]. Thus, there are different strategies for the autonomous acquisition of place cells, each of which is, however, tightly coupled to the behavioral paradigm in question.

Every type of navigation requires the acquisition and exploitation of a world model. Starting with the work of Tolman, two schools of thought about the type of such a world model have been distinguished [88]. The “stimulus-response school” states that navigation implies the direct mapping between sensory stimuli and motor responses. The “cognitive map school” postulates the formation of an internal representation of the external world. Thus, the two schools are not contradictory, but the latter does rather conceptualize the type of intermediate internal computations and representations which could be particularly useful for navigation. The question arises, however, what kind of representation would be particularly useful for what type of navigation.

On one hand, an egocentric representation of the world as in the stimulus-response approach allows a simple mapping between orientation dependent sensor readings and relative actions. On the other hand it does not provide any notion of environmental topology as an allocentric representation does. Navigation based on the latter, however, relies on global (allocentric) actions, like going south, which in turn have to be translated back to an egocentric representation in order to become meaningful. Thus while an egocentric representation seems more appropriate for the reactive control of an organism, the allocentric representation has its strengths in the cognitive control. There is no a priori reason why not both representations could be in use simultaneously. This is in contrast to studies modeling the formation of place cells and emphasizing their direct role in goal oriented behavior [8, 18] while neglecting any direct egocentric stimulus-response

mappings. Even more, experimental studies suggest that the hippocampus plays an important role in spatial learning while it is not required to exploit navigational skills which have been acquired before a lesion or inactivation of the hippocampus [7, 61]. Thus, we hypothesize that place cells assist an egocentric stimulus-response based navigation system by providing allocentric information rather than playing an active role in navigation itself. For an egocentric navigation system, for example, it is impossible to identify a location while approaching from different directions. Access to this kind of information, however, allows to split a complete navigation task into several subtasks such that learning periods can be shortened significantly.

Furthermore, we believe that the egocentric representations of the environment as given by the temporal population code are not only interesting in terms of the formation of allocentric representations of space, i.e. place fields. They might also play a vital role in egocentric navigational strategies such as visual guidance for which the property of bounded invariance would allow for both, specificity and tolerance in cue detection.

Chapter 6

Conclusion

The goal of the present dissertation was to study the coding of information in sensory as well as motor systems in the context of a behaving system. The original motivation for this work came from a very general question in theoretical computational neuroscience, namely, to what extent can artificial neural networks approximate real biological systems, if they are not embedded in the behavioral feedback loop like biological organisms interacting with their environment. Any type of coding depends strongly on the properties of the information to be encoded as well as the requirements of the observer reading and interpreting the code. Thus, in regard of the behavioral loop, the interaction of an agent with its environment can put severe constraints on the type of coding used, both at the sensory and the motor end of the system. Therefore, the validity and relevance of a coding strategy may not be fully established unless it is tested and analyzed in the context of a complete behaving system.

In the first part of the thesis, the temporal population code was introduced as a new proposal of how information is encoded and processed in neural systems. Starting from the initial question whether and how the brain may exploit time as an additional coding dimension, the temporal population code goes one step further in neglecting the spatial dimensions and suggesting a purely temporal coding strategy. It is probably safe to say that the brain would not employ such an extreme strategy throughout, but rather uses a mixed approach depending on the needs of the different subsystems in question. One of the aims of this thesis was, however, to demonstrate the potential advantages of temporal coding in combination with non-topological statistics as well as the latter by itself. In particular,

it was shown that the temporal population code exhibits properties which are of critical importance for behavioral systems which interact in real time with noisy environments. The speed at which information becomes encoded is very fast. In addition, the encoding progresses incrementally such that information can be read by subsequent stages of processing before the encoding process has actually completed. This allows for the concurrent processing of information across different stages which can be of great importance for behaving systems engaged in time critical tasks. Using an enhanced version of the encoding network, reflecting known properties of the primary visual cortex, it was also shown that the temporal population code is well suited for those pattern recognition tasks mammals excel at, e.g. to reliably classify large stimulus sets of increased complexity. The system can cope with significant within class variability as found in the standard pattern recognition benchmark database MNIST for which it achieves classification results close to 95%. This falls short of other highly optimized pattern classifiers. Given the fact, however, that the system considered does not use any learning but is merely transforming the spatial input patterns into a different representation, this performance is rather remarkable.

An important question regarding coding strategies is how information becomes encoded and how it is decoded. The model of a cortical network considered in the first part of this thesis is responsible for the generation of the temporal population code. This network, in particular in its enhanced version, reflects some basic properties of primary visual cortex such as the orientation and spatial frequency selectivity of the individual neurons and the non-isotropic connectivity between neurons depending on their individual feature selectivity. More importantly, however, the model assigns a clear role to the extensive lateral connectivity found within cortical structures. This is in contrast to most hierarchical models of pattern recognition which mainly focus on feed-forward processing of information. For the cortical network discussed, the lateral connections and more precisely the transmission properties of these connections play a key role in the coding process. The temporal population code is based on the coordinated temporal activity patterns of a large population of neurons. Thus, no global reference signal is needed, to which the spike timing of different cells needs to be adjusted in order to encode information. The lateral connectivity between the neurons adjusts the relative spike timing of different

cell pairs, and the statistics of these relative spike timings constitutes the code. More generally spoken, since it is probably more reasonable to think of temporal coding in terms of relative spike timing between cells rather than global spike timing, lateral recurrent interactions are inevitable. Recurrent interactions lead to non-trivial network dynamics, meaning that the temporal activation patterns of neurons are not merely reflecting the temporal structure of their input. The vast recurrent connectivity in biological neural systems does not necessarily imply that the brain must use temporal coding strategies, however, it does not rule it out either.

In the studies presented in this thesis, the temporal population code was read out using purely statistical methods only. However, we studied a more neural approach towards this problem in the context of a different project [41], where we investigated a recent proposal for the processing of continuous temporal streams of information, the *Liquid State Machine* [49]. This model is based on the structure and dynamics of cortical micro-circuits which essentially project the information to be decoded into a high dimensional space, where it can be classified using simple perceptron-like classifiers. Employing the original proposal with a liquid that is initialized randomly before stimulus presentation results in a moderate performance. Based on an analysis of the liquid's internal states, we explored further initialization strategies. Whereas a deterministically initialized liquid results in the best performance, we find that in case the liquid is never reset, i.e. it continuously processes the sequence of stimuli, the classification performance is greatly hampered by the mixing of information from past and present stimuli. This problem of the mixing of temporally segregated information is not specific to this particular method but relates to a general problem that any circuit that processes continuous streams of temporal information needs to solve. Considering the biological realism of the encoding and decoding stage, this study suggests that the brain solves the problem of temporal mixing by applying reset signals at stimulus onset. In fact, the initial sharp onset response observed in the temporal population code would be particularly well suited to convey this type of signal indicating the onset of a stimulus to trigger the read out system.

As already discussed, the assessment of any coding strategy is strongly dependent on its range of applications. However, the uniformity of cortical anatomy suggests affinities between the coding of information of different modalities in cortical structures. Therefore, the objective of the second

part of this dissertation was to investigate whether the coding principles studied would generalize to other modalities. In particular, we were interested, to what extent the concept of non-topological codes would be advantageous in the motor system. In a simple model of sensori-motor control, we studied the general problem of how an autonomous system could detect the presence of different behavioral options. It was shown, that applying a simple non-topological statistical measure to the motor system allowed to reliably detect these behaviorally ambiguous situations. Due to the invariance properties of non-topological codes, the detection was invariant to the detailed sensory manifestation of these situations. Thus the same coding principle can be used in the motor system. It must be emphasized, however, that in this context, the motor system is rather a high level perceptual system than an output system. The nature of non-topological codes and in particular its invariance implies that they are not very well suited for the encoding of motor commands. This system, however, shows very nicely, that non-topological codes can coexist with other codes in the same physical substrate without interference. Thus, the motor system uses a population vector coding strategy to encode the required movements of the robot to fulfill the sensori-motor task on top of which the non-topological system can determine the behavioral relevance of situations.

In the last part of this thesis, the bounded invariance properties of the temporal population code were investigated. A representation of a stimulus is bounded invariant to a particular transformation, when its representation does not change for small transformations, however, it does change for larger transformations. An alternative view is, that the representation is topology preserving, i.e. similar/dissimilar stimuli (in terms of the transformation in question) have similar/dissimilar representations. It was shown that the temporal population code has this property in that it preserves an intuitive notion of visual similarity. Specifically, it was shown that it is bounded invariant to visual stimuli which undergo transformations like zooming or rotation in depth. Considering an autonomous agent, its visual percepts of the environment will undergo a combination of these two transformation while moving around. Thus, small movements will lead to similar representations of the visual stimuli while larger displacements will lead to completely different internal representations. It was shown, that this effect can be exploited to construct place cells, i.e. cells which are

selectively active depending on the agents position within the environment.

Interestingly, the concept of bounded invariance is closely related to a different recently proposed coding strategy which postulates that a common principle for the internal representation of sensory information is that these representations should vary on a slow time scale only [14, 39, 98]. An agent moving continuously in space and time will perceive its environment in a mostly continuous manner. Thus, a representation of this environment with bounded invariance will change at the time scale at which the global percept is changing rather than, for example, at the rate at which a single photo-receptor is responding to its stimulation. Thus, in the context of a behaving agent, a bounded invariant representation also conforms to the concept of “slowness”, i.e. it changes on the behavioral rather than a neural time-scale. In a preliminary study, we investigated to what extent this argumentation holds by studying a system which uses optimization techniques to construct representations from multi-modal sensory information which conform to the proposed “slowness” criterion. Indeed, we found that the cells emerging after several such optimization stages, have developed response properties similar to place cells found in rat hippocampus. Thus, it will be the objective of future work to elaborate on these preliminary results in order to detail what the conceptual similarities and differences between the temporal population code, specifically its various invariance properties, and the recently proposed optimization approaches are.

Appendix A

The `wSim` simulation environment

A.1 Motivation - yet another neural simulator?

There are many ways to simulate neural networks. The different tools may range from very general purpose tools such as Matlab or C++ to very specialized programs such as *Neuron* [81] or *Genesis* [16] which put a strong emphasis on biophysically detailed neuronal models. The individual choice depends very much on the specific needs of a project.

In principle, today's computers and the tools that come with it are particularly badly suited to simulate neural networks. The greatest mismatch arises due to the two most prominent properties of neural systems, their structural diversity and their massively parallel operation. Modern computers and algorithms, however, are optimized for serial execution of instructions and homogeneous data structures.

Unfortunately, attacking the problem regarding parallel computation with clusters of computers or multi-processor systems is in general not very promising because the communication between the systems becomes the limiting factor. Thus, for very specific problems, where a neural network can be split into different modules which require minimal communication between each other, such an approach might be applicable. The distribution of massively inter-connected networks, however, will generate a large amount of communication overhead. Thus, for now, the only way to

“tackle” the problem of massively parallel computation is to optimize the serial implementation as much as possible in order to achieve an optimal performance on a single processor.

The structural diversity of neural networks manifests itself at different levels. First, there are a variety of neural cell and synapse types one would possibly like to use. Second, connection patterns between different cell groups might be sparse or all-to-all, random or structured. Thus, in both cases, one faces a trade-off between using customized code which is very fast but unportable and hard to reuse, as opposed to follow a more general approach which, however, might lead to poor performance.

In summary, there is a need for a solution, which is maximally flexible while conceding a minimum of compromises in terms of performance. Therefore, the strategy in the design of the `wSim` simulation environment was to provide an elaborate C++ library, allowing for maximal flexibility while relying on heavy use of *templates*, admitting for extensive optimizations at compile-time. In the following, some critical design issues concerning the main `wSim` library, as well as the different components of the complete simulation environment will be discussed.

A.2 Design strategies

The main design considerations responsible for both the flexibility of the library as well as its highly optimized performance will be discussed in the following sections.

A.2.1 Polymorphism and templates

The structural diversity of neural networks suggests a concept which was mainly introduced with object oriented programming, i.e. *polymorphism* (see [82] for a general introduction). Run-time polymorphism refers to a programming language’s ability to process objects differently depending on the object type, whereas the decisions about which code to execute are taken at run-time. Thus, a library for neural networks could use this mechanism to selectively execute code depending on the type of cell or synapse currently used. Unfortunately, the additional machinery required for this type of flexibility can have a severe effect on run-time performance,

specifically if this machinery has to be invoked for each cell or synapse. This problem can be circumvented by choosing the smallest object to be for example a group of cells instead of single cells. This, however, is by far not as elegant and makes the library less intuitive for the user. Furthermore, such a design of the library would have implications on the data alignment of the library which is discussed in the next section.

An alternative to run-time polymorphism are *templates* which provide compile-time polymorphism, i.e. decisions about which code to execute are performed at compile-time. Thus templates may cause an increase in compilation time while having no impact on the run-time performance of a library. In addition, using templates allows for a *policy-based* design of the library which guarantees for maximal flexibility and scalability [5]. In essence, this means that the functionality of the library is split into orthogonal components, i.e. mutually independent components, which then can be combined in a single instance at compile-time through templates. This yields a very diverse and versatile functionality which is much superior to other mechanisms such as multiple inheritance.

A.2.2 Data alignment and locality of information

Another important issue in simulating neural networks is the alignment of data. Usually, networks become rather large and therefore require a lot of memory to store the state variables of the individual cells and synapses. An important component in modern computers is the cache, a special type of memory which is directly located on the processor, providing extremely fast access to data. Compared to the normal memory, however, the cache is usually a lot smaller, such that only a limited amount of consecutive data can be stored. Thus, if a program requires access to data which is not in the cache, the main memory has to be consulted, which slows down processing significantly. Therefore, it is advisable to organize the state variables such that the ones that share a common computation are located near each other in memory such that it becomes very likely to have them cached simultaneously.

Since a common property of neural networks is that computations are based on local information only, i.e. a cell only knows about its own potential, the state variables of a single entity should be grouped in memory. This is in contrast to another approach where one would rather group the

state variables of the same type across cells. Regarding the question about the smallest object raised in the previous section, this implies that each cell forms a single object rather than a group of cells being the smallest entity. Thus, both, in terms of library design (intuitive, data encapsulation) and performance (compile-time optimizations, data alignment), the strategy of defining the single entities, cells and synapses, as individual objects is well justified and was pursued in the development of the `wSim` simulation environment.

A.3 `wSim` components

A.3.1 Library

The library is the main component of the simulation environment providing the basic functionality to define and build neural networks. Each network is organized in hierarchy in which *modules* form the basic structural unit. A module is a container which can hold other modules, groups of cells (`CellGrp`) and groups of synapses (`SynGrp`).

The library provides the founding to define cell and synapse types used to instantiate `CellGrp`'s as well as `SynGrp`'s. A cell type, for example, is defined by declaring a new class which derives from the base cell type `WCellType<>` provided by the library, i.e.

```
#include <wSim.hpp>
```

```
class MyCell : public WCellType<float> { /* ... */ };
```

where the template argument `<float>` declares the output of the cell to be a single precision floating point number. Similarly, synapse types are derived from a base synapse type `WSynType<>`

```
class MySynapse : public WSynType<AffCellType,
                                EffCellType> {
    /* ... */
};
```

where `AffCellType` and `EffCellType` are the afferent and efferent cell types of `MySynapse`. For both, cells and synapses, state variables are defined as members and the computations performed on these variables are

defined as member functions. The following example shows how this might look for a very simple linear unit:

```
class MyCell : public WCellType<float> {
public:
    // two state variables, activation and input
    float act, input;

    // this function is called for each time step
    float update() {
        act=input;
        input=0;
        return act;
    }
};
```

For a more detailed and elaborate example, please see section A.4.

To instantiate a group of cells of type `MyCell`, one creates an object of type `WCellGrp<MyCell>` i.e.

```
new WCellGrp<MyCell>(module, "MyCell", 20, 20);
```

where the first argument is a pointer to a module to which this `CellGrp` should belong, the second argument is a string with the objects name, and the last two arguments give the size of the group, i.e. 20×20 cells. Similarly, a `SynGrp` is created by the following instruction:

```
new WSynGrp<MySynapse>(afferent, efferent);
```

where `afferent` and `efferent` are pointers to the pre- and postsynaptic `CellGrp`'s respectively.

Thus, the library by itself provides the basic functionality to define and build a hierarchical network of different cell and synapse types whereas the different objects can be grouped in different modules. Running such a network thereafter, is a rather straight forward task. Each module contains a cycle function which updates all the objects and sub-modules. Thus, by calling the cycle function of the root module, also called the process, all objects are updated recursively in an order given by the hierarchy.

A.3.2 Engine

The simulation environment provides two different engines to run the process. A very simple and light-weight non-interactive engine runs at the console. A more elaborate engine provides a graphical user-interface (GUI) which allows direct interaction with the running process as well as the on-line inspection of different state variables. Both engines dynamically load the process as compiled binary code. The console-based engine, thereafter, repeatedly calls the process' cycle function until it gets terminated by the user. The purpose of this engine is mainly to run automated large scale simulation in the background where no user interaction is required.

The GUI engine's primary use is to simulate processes in the foreground providing the user different ways of interaction. The basic functionality of this engine is:

- starting, stopping and initializing a process
- on-line manipulation of network parameters
- visualization of state variables (see fig. A.1)
- selective activation and deactivation of modules/objects
- exporting state variables to Matlab

The visualization of state variables is kept at a minimal level. There are space plots of cell as well as synapse groups. Furthermore, each state variable of different cells/synapses or groups of those can be inspected in time plots or histograms. For more elaborate data visualization, the engine provides the possibility to export data to Matlab which provides a much richer and more flexible graphics facility.

A.3.3 Plugins

As a special type of module, the library also defines devices. These are plugins which allow the interaction with external devices such as robots or cameras. At the present state, four different devices are defined: `wKhepera`, `wVideo`, `wKleopatra`, `wAudio`.

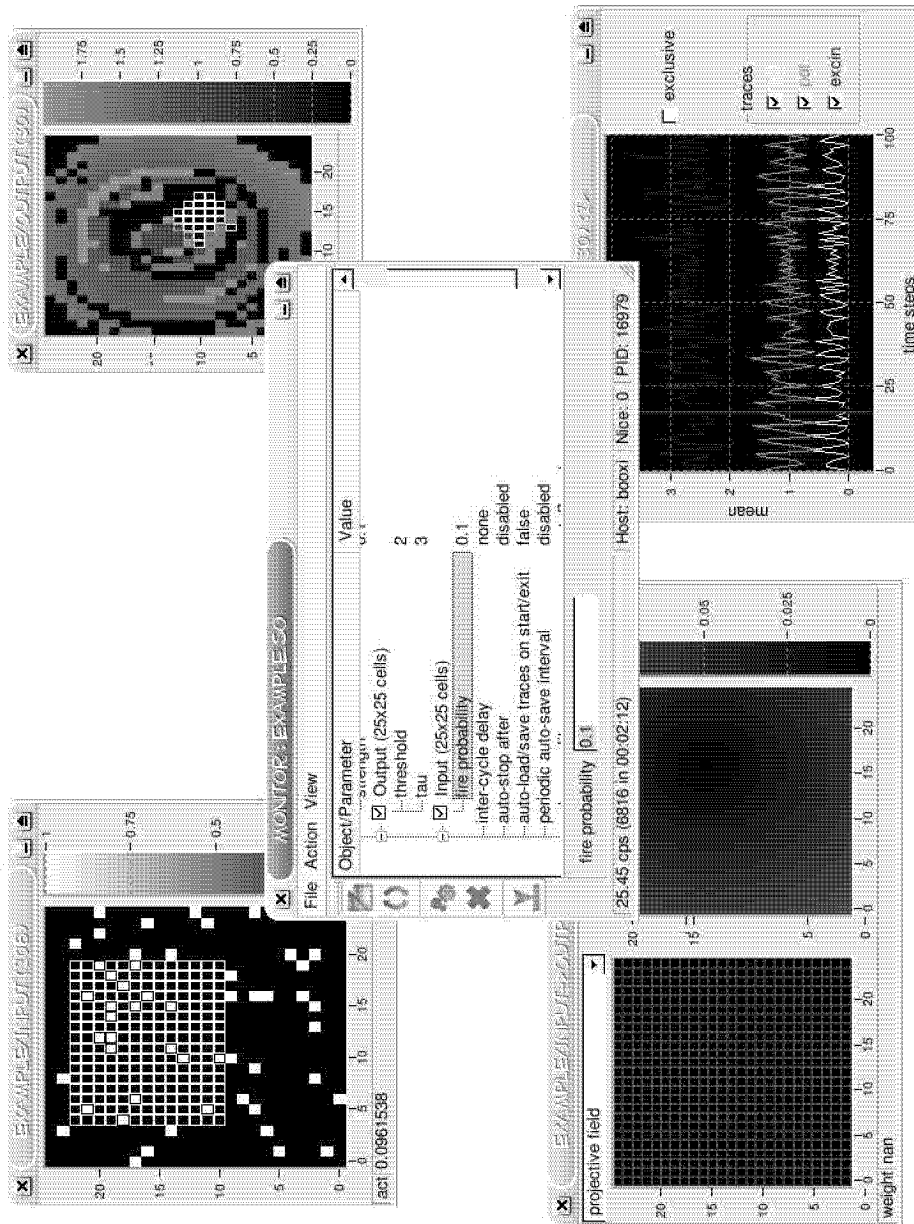


Figure A.1: The GUI engine of the wSim simulation environment consists of a main window (center), which allows to interact with different modules as well as objects, i.e. CellGrps and SynGrps. Different monitors allow to inspect traces of objects online. The two windows on top show group plots of two CellGrps. The monitor in the lower left corner shows the traces of a SynGrp and its arborization. The lower right corner contains a time plot, which allows to investigate the evolution of traces over time for both, CellGrps and SynGrps. In addition, selections of single or groups of cells or synapses can be visualized using an intuitive drag-n-drop interface.

The `wKhepera` device provides an interface to the mobile robot Khepera (fig. 4.1a). Special `CellGrp`'s provide the facility to drive the robots wheels, as well as to read the different sensory values from the eight IR-sensors arranged around the robots body. The `wVideo` device provides access to CCD cameras which are often mounted on top of the Khepera robot. A special video `CellGrp` reads this images and provides them to the rest of the simulation.

The `wKleopatra` device is a virtual approximation of the Khepera robot, providing the same functionality. This robot moves in a virtual environment which can be specified by a scenery description file. The environment is rendered in OpenGL as seen from a virtual camera mounted on top of the robot. Thus, the `wKleopatra` device provides both, access to the virtual robot as well as the virtual camera. A main goal in the design of these interfaces was to provide a common interface for both real-world as well as virtual robot, such that the same simulation can be performed in the virtual as well as a real environment, simply by choosing a different back-end. This is very convenient, because often, large scale experiments are performed preferably in a virtual environment, whereas the real-world experiment is necessary to show that a specific model can cope with the real world. This strategy, for example, was successfully applied in the work described in chapter 4.

Last but not least, the `wAudio` provides access to auditory information, i.e. it interfaces to the soundcard of a computer and provides the power spectrum of a user-defined input source.

A.4 Example process

The following example illustrates how a complete process with two `CellGrp`'s of cell types `InputCell` and `OutputCell` and a connecting `SynGrp` of type `MySynapse` is built. The header file contains the declarations of the different cell and synapse types as well as the process. In the source file, the process is then constructed by instantiating the `CellGrp`'s and the `SynGrp` within the constructor of the process.

```

/*****

Header File for wSim process

Project      : example
Author       : rwyss
Started      : Tue Aug 5 10:05:48 2003
Description  :
Change log   :

*****/

#include <wSim.hpp>

/*****/
/*****/ Cell Definitions *****/
/*****/

// a randomly spiking cell
class InputCell : public WCellType<bool,1> {
public:

    float  act;
    static float fireProb;

// this function is called for each cell
// at each time step; return true if a spike
// is emitted, false otherwise
    bool update() {
        act=randu(<fireProb;
        return act>0;
    }

// this function is called once, when a group of cells of
// type 'MyCell' is created; thus, specify which traces
// (pot, excin ...) or static parameters (thresh, tau ...)
// should be accesible within the GUI

```

```
static void define() {
    WTrace::add(&InputCell::act , "activity", "act",
               false, QColor(255,255,255));
    WParam::add(&fireProb, false, "fire probability", 0.1);
}

static const QString getDescription() {
    return "random spiking cell";
}
};

class OutputCell : public WCellType<bool,0> {
public:

// an integrate and fire cell
    float act, pot, excin;
    static float thresh, tau;

// this function is called for each cell
// at each time step; return true if a spike
// is emitted, false otherwise
    bool update() {
        pot+=1/tau*(excin-pot);
        excin=0; // reset
        if (pot>thresh) {
            pot=0;
            act=1;
            return true;
        } else {
            act=0;
            return false;
        }
    }

// this function is called for each cell
// when the process is reset or parameters are
// changed; v is 3D vector indicating the cells
```

```

// position within the cell group
void init(WVector v) {
    pot=excin=0; // reset all values to zero
}

// this function is called once, when a group of cells of
// type 'MyCell' is created; thus, specify which traces
// (pot, excin ...) or static parameters (thresh, tau ...)
// should be accesible within the GUI
static void define() {
    WTrace::add(&OutputCell::act ,"activity","act",
                false,QColor(255,255,255));
    WTrace::add(&OutputCell::pot ,"potential","pot");
    WTrace::add(&OutputCell::excin ,"exc. input","excin");
    WParam::add(&thresh,false,"threshold",2);
    WParam::add(&tau,false,"tau",3);
}

static const QString getDescription() {
    return "integrate & fire cell";
}
};

/*****
/***** Synapse Definitions *****/
/*****/

class MySynapse : public WSynType<InputCell,OutputCell,
                                FwdMask> {
public:
    static float strength;
    float weight;

// this function is called (d+1) time-steps after the
// presynaptic cell fired, where d is the delay set by
// the function void setDelay(int d);
// a: presynaptic, b: postsynaptic

```

```

void update(InputCell &a, OutputCell &e) {
    e.excin+=weight;
}

// see above; v1: presynaptic, v2: postsynaptic
void init(WVector v1,WVector v2) {
    setDelay(0);
    // f.ex. set weight to the topographic distance
    // between pre- and postsynaptic cell
    weight=strength*length(v1-v2);
}

// see above
static void define() {
    WTrace::add(&MySynapse::weight,"weight");
    WParam::add(&strength,true,"strength",0.1);
}

static const QString getDescription() {
    return "excitatory all-to-all connection";
}
};

/*****
/***** Process Definition *****/
/*****/

class MyProcess : public WProcess {
public:
    MyProcess();

protected:
    WCellGrp<InputCell> *input;
    WCellGrp<OutputCell> *output;

    WSynGrp<MySynapse> *synapse;

```

};

```

/*****

Source File for wSim process

Project      : example
Author       : rwyss
Started      : Tue Aug 5 10:05:48 2003
Depends      : example.hpp
Description  :
Change log   :

*****/

#include "example.hpp"

float InputCell::fireProb;
float OutputCell::thresh, OutputCell::tau;
float MySynapse::strength;

MyProcess::MyProcess() : WProcess("example") {

// constructing two cell groups, each of size 25x25
  input  = new WCellGrp<InputCell>(this,"Input",25,25);
  output = new WCellGrp<OutputCell>(this,"Output",25,25);

// connecting the two cell groups
  synapse = new WSynGrp<MySynapse>(input,output);

}

/*****/
/** DO NOT CHANGE ANYTHING BEYOND THIS LINE */
/*****/

// this function is called from wSimMonitor/wSim
extern "C" WProcess *createProcess() {
// creates the process and returns a pointer to it

```

```
    return (WProcess*) new MyProcess();  
}
```


Bibliography

- [1] M. Abeles. *Local cortical circuits. An electrophysiological study.* Springer, Berlin, 1982.
- [2] E. Adrian. *The Basis of Sensation: The Action of the Sense Organs.* W.W. Norton, New York, 1928.
- [3] E. Adrian. *The Mechanism of Nervous Action: Electrical Studies of the Neurone.* Univ. Pennsylvania Press, Philadelphia, 1932.
- [4] H. Agmon-Snir and I. Segev. Signal delay and input synchronization in passive dendritic structures. *J. Neurophysiol.*, 70:2066–85, 1993.
- [5] A. Alexandrescu. *Modern C++ Design: Generic Programming and Design Patterns Applied.* Addison Wesley, 1st edition, February 2001.
- [6] C. Allen and C.F Stevens. An evaluation of causes for unreliability of synaptic transmission. *Proc. Natl. Acad. Sci. USA*, 91:943–52, 1994.
- [7] S.H. Alyan, R. Jander, and P.J. Best. Hippocampectomized rats can use a constellation of landmarks to recognize a place. *Brain Res*, 876(1-2):225–37, 2000.
- [8] A. Arleo and W. Gerstner. Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity. *Biol Cybern.*, 83(3):287–99, 2000.
- [9] H.R. Arnoldi, K. Engelmeier, and W. Brauer. Translation-invariant pattern recognition based on synfire chains. *Biol. Cyber.*, 80:433–47, 1999.

- [10] P. Auer, H. Burgsteiner, and W. Maass. Reducing communication for distributed learning in neural network. In José R. Dorronsoro, editor, *Proc. of the International Conference on Artificial Neural Networks – ICANN 2002*, pages 123–128. Springer, 2002.
- [11] W. Bair. Spike timing in the mammalian visual system. *Curr. Opin. Neurobiol.*, 9(4):447–53, 1999.
- [12] H. Barlow. Single units and sensation: A neuron doctrine for preceptual psychology. *Perception*, 1:371–94, 1972.
- [13] A. Benucci, P.F.M.J. Verschure, and P. König. The impact of network dynamics on single cell activity. *Soc. Neurosc. Abstracts.*, in press, 2002.
- [14] P. Berkes and L. Wiskott. Applying slow feature analysis to image sequences yields a rich repertoire of complex cell properties. In José R. Dorronsoro, editor, *Proc. Intl. Conf. on Artificial Neural Networks - ICANN'02*, Lecture Notes in Computer Science, pages 81–6. Springer, 2002.
- [15] W.H. Bosking, Y. Zhang, B. Schofield, and D. Fitzpatrick. Orientation selectivity and the arrangement of horizontal connections in tree shrew striate cortex. *J. Neurosci.*, 17(6):2112–27, 1997.
- [16] J.M. Bower, D.Beeman, and M. Hucka. The GENESIS simulation system. In M.A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 475–478. MIT Press, Cambridge, MA, 2nd edition, 2003.
- [17] D.V. Buonomano and M. Merzenich. A neural network model of temporal code generation and position-invariant pattern recognition. *Neural Comput.*, 11:103–16, 1999.
- [18] N. Burgess, J.G. Donnett, H.J. Jeffrey, and J. O’Keefe. Robotic and neuronal simulation of the hippocampus and rat navigation. *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 352(1360):1535–43, 1997.

- [19] A.F. Carpenter, A.P. Georgopoulos, and G. Pellizzer. Motor cortical encoding of serial order in a context-recall task. *Science*, 283(5408):1752–7, 1999.
- [20] S.M. Crook, G.B. Ermentrout, M.C. Vanier, and J.M. Bower. The role of axonal delay in the synchronization of networks of coupled cortical oscillators. *J. Comput. Neurosci.*, 4:161–72, 1997.
- [21] R. de Ruyter van Steveninck, G.D. Lewen, S.P. Strong, R. Koberle, and W. Bialek. Reproducibility and variability in neural spike trains. *Science*, 275(5307):1805–8, 1997.
- [22] R.J. Douglas and K.A.C. Martin. Neocortex. In G.M. Shepherd, editor, *The Synaptic Organization of the Brain.*, pages 389–438. Oxford University Press,, 1990.
- [23] A.K. Engel, P. König, A.K. Kreiter, and W. Singer. Interhemispheric synchronization of oscillatory neural responses in cat visual cortex. *Science*, 252:1177–9, 1991.
- [24] W.J. Freeman. *Mass Action In The Nervous System*. Academic Press, New York, 1975.
- [25] I. Fujita, K. Tanaka, M. Ito, and K. Cheng. Columns for visual features of objects in monkey inferotemporal cortex. *Nature*, 360:343–6, 1992.
- [26] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybernetics*, 36:193–202, 1980.
- [27] A. Gail, H.J. Brinksmeier, and R. Eckhorn. Contour decouples gamma activity across texture representation in monkey striate cortex. *Cerebral Cortex.*, 10(9):840–50, 2000.
- [28] J.L. Gardner and S.G. Lisberger. Serial linkage of target selection for orienting and tracking eye movements. *Nat Neurosci.*, 5(9):892–9, 2002.

- [29] A.P. Georgopoulos, A.B. Schwartz, and R.E. Kettner. Neural population coding of movement direction. *Science*, 233:1416–9, 1986.
- [30] W. Gerstner. Spiking neurons. In W. Maas and C.M. Bishop, editors, *Pulsed Neural Networks.*, chapter 1, pages 3–53. MIT Press, Cambridge, MA, 1998.
- [31] C. Golgi. The neuron doctrine – theory and facts. In *Nobel Lectures: Physiology or Medicine 1901-1921.*, pages 189–217. Elsevier, Amsterdam, 1967.
- [32] A. Hernandez, A. Zainos, and R. Romo. Temporal evolution of a decision-making process in medial premotor cortex. *Neuron*, 33(6):842–4, 2002.
- [33] J.J. Hopfield. Neural networks and physical systems with emergent collective computational properties. *Proc. Natl. Acad. Sci. USA*, 79:2554–8, 1982.
- [34] J.J. Hopfield. Pattern recognition computation using action potential timing for stimulus representations. *Nature*, 376:33–6, 1995.
- [35] D. Hubel and T. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *J. Physiol.*, 160:106–154, 1962.
- [36] A. Hughes. A schematic eye for the rat. *Visual Res.*, 19:569–88, 1977.
- [37] M. Ito, H. Tamura, I. Fujita, and K. Tanaka. Size and position invariance of neuronal responses in monkey inferotemporal cortex. *J. Neurophysiol.*, 73:218–26, 1995.
- [38] E.R. Kandel, J.H. Schwartz, and T.M. Jessell. *Principles of Neural Science*. McGraw-Hill, 4th edition, 2000.
- [39] C. Kayser, W. Einhuser, O. Dümmer, P. König P, and K. Körding. Extracting slow subspaces from natural videos leads to complex cells. In *Proc. Intl. Conf. on Artificial Neural Networks - ICANN’01*, Lecture Notes in Computer Science 2130, page 1075 ff. Springer, 2001.

- [40] J. Knierim, H. Kudrimoti, and B. McNaughton. Place cells, head direction cells, and the learning of landmark stability. *J. Neurosci.*, 15:1648–59, 1995.
- [41] P. Knüsel, R. Wyss, P. König, and P. Verschure. Decoding a temporal population code. *submitted for publication*.
- [42] E. Kohler, C. Keysers, M.A. Umiltà, L. Fogassi, V. Gallese, and G. Rizzolatti. Hearing sounds, understanding actions: action representation in mirror neurons. *Science*, 297:846–8, 2002.
- [43] P. König and A.K. Engel. Correlated firing in sensory-motor systems. *Curr. Opin. Neurobiol.*, 5:511–9, 1995.
- [44] P. König, A.K. Engel, P.R. Roelfsema, and W. Singer. How precise is neural synchronization? *Neural Comput.*, 7:469–85, 1995.
- [45] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–324, Nov. 1998.
- [46] M.I. Leon and M.N. Shadlen. Exploring the neurophysiology of decisions. *Neuron*, 21:669–72, 1998.
- [47] N.K. Logothetis and D.L. Sheinberg. Visual object recognition. *Annu. Rev. Neurosci.*, 19:577–621, 1996.
- [48] W. Maass. Fast sigmoidal networks via spiking neurons. *Neural Comput.*, 9(2):279–304, 1997.
- [49] W. Maass, T. Natschlager, and H. Markram. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.*, 14(11):2531–60, 2002.
- [50] C.K. Machens, M.B. Stemmler, P. Prinz, R. Krahe, B. Ronacher, and A.V. Herz. Representation of acoustic communication signals by insect auditory receptor neurons. *J. Neurosci.*, 21(9):3215–27, 2001.
- [51] K. MacLeod, A. Backer, and G. Laurent. Who reads temporal information contained across synchronized and oscillatory spike trains? *Nature*, 395(6703):693–8, 1998.

- [52] J.W. McClurkin, T.J. Gawne, L.M. Optican, and B.J. Richmond. Lateral geniculate neurons in behaving primates. II. Encoding of visual information in the temporal shape of the response. *J. Neurophysiol.*, 66(3):794–808, 1991.
- [53] P.M. Milner. A model for visual shape recognition. *Psychol. Rev.*, 81:521–35, 1974.
- [54] U. Mitzdorf and W. Singer. Prominent excitatory pathways in the cat visual cortex (A 17 and A 18): a current source density analysis of electrically evoked potentials. *Exp. Brain. Res.*, 33:371–94, 1978.
- [55] F. Mondada and P.F.M.J. Verschure. Modeling system-environment interaction: The complementary roles of simulations and real world artifacts. In H. Bersini, S. Goss, G. Nicolis, and R. Dagonnier, editors, *Proceedings of the Second European Conference on Artificial Life*, pages 808–817, Cambridge, MA, 1993. The MIT press.
- [56] E. Naito, P.E. Roland, and H.H. Ehrson. I feel my hand moving: a new role of the primary motor cortex in somatic perception of limb movement. *Neuron*, 36(5):979–88, 2002.
- [57] L.G. Nowak, M.H.J. Munk, J.I. Nelson, and A.C. Bullier. Structural basis of cortical synchronization, 1. Three types of interhemispheric coupling. *J. Neurophysiol.*, 74:2379–400, 1995.
- [58] J. O’Keefe and N. Burgess. Geometric determinants of the place fields of hippocampal neurons. *Nature*, 381(6581):425–8, 1996.
- [59] J. O’Keefe, N. Burgess, J.G. Donnett, K.J. Jeffrey, and E.A. Maguire. Place cells, navigational accuracy, and the human hippocampus. *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 353(1373):1333–40, 1998.
- [60] J. O’Keefe and L. Nadel. *The hippocampus as a cognitive map*. Clarendon Press, Oxford, 1987.
- [61] M.G. Packard and J.L. McGaugh. Inactivation of hippocampus or caudate nucleus with lidocaine differentially affects expression of place and response learning. *Neurobiol. Learn. Mem.*, 65(1):65–72, 1996.

- [62] D.I. Perrett and M.W. Oram. Neurophysiology of shape processing. *Imaging Vis. Comput.*, 11:317–33, 1993.
- [63] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, Cambridge, New York, Port Chester, Melbourne, Sydney, 1988.
- [64] G. Quirk, R. Muller, and R. Kubie. The firing of hippocampal place cells in the dark depends on the rat’s recent experience. *J. Neurosci.*, 10:2008–17, 1995.
- [65] S. Ramón y Cajal. *Histology*. Wood, Baltimore, 10th edition, 1933.
- [66] B.J. Richmond and L.M. Optican. Temporal encoding of two-dimensional patterns by single units in primate inferior temporal cortex. II Quantification of response waveform. *J. Neurophysiol.*, 57(1):147–61, 1987.
- [67] B.J. Richmond and L.M. Optican. Temporal encoding of two-dimensional patterns by single units in primate primary visual cortex. II. Information transmission. *J. Neurophysiol.*, 64(2):370–80, 1990.
- [68] F. Rieke, D. Warland, R. de Ruyter van Steveninck, and W. Bialek. *Spikes, exploring the neural code*. MIT Press, Massachusetts Institute of Technology, 1997.
- [69] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nat. Neurosci.*, 2(11):1019–25, 1999.
- [70] G. Rizzolatti and G. Luppino. The cortical motor system. *Neuron*, 31(6):889–901, 2001.
- [71] E.T. Rolls. Functions of the primate temporal lobe cortical visual areas in invariant visual object and face recognition. *Neuron*, 27(2):205–18, 2000.
- [72] E.T. Rolls, A. Treves, and M.J. Tovee. The representational capacity of the distributed encoding of information provided by populations of neurons in primate temporal visual cortex. *Exp. Brain Res.*, 114(1):149–62, 1997.

- [73] D. Rumelhart and J. McClelland. *Parallel Distributed Processing*. MIT Press, Cambridge, MA, 1986.
- [74] E. Salinas and L.F. Abbott. Vector reconstruction from firing rates. *J. Comput. Neurosci.*, 1:89–107, 1994.
- [75] E. Salinas and R. Romo. Conversion of sensory signals into motor commands in primary motor cortex. *J. Neurosci.*, 18(1):499–511, 1998.
- [76] M.T. Schmolesky, Y. Wang, D.P. Hanes, K.G. Thompson, S. Leutgeb, J.D. Schall, and A.G. Leventhal. Signal timing across the macaque visual system. *J. Neurophysiol.*, 79(6):3272–8, 1998.
- [77] M.N. Shadlen, K.H. Britten, W.T. Newsome, and J.A. Movshon. A computational analysis of the relationship between neuronal and behavioral responses to visual motion. *J. Neurosci.*, 16(4):1486–510, 1996.
- [78] W. Singer. Neuronal synchrony: a versatile code for the definition of relations? *Neuron*, 24(1):49–65, 111–25, 1999.
- [79] W. Singer. Time as coding space? *Curr. Opin. Neurobiol.*, 9(2):189–94, 1999.
- [80] W. Singer and C.M. Gray. Visual feature integration and the temporal correlation hypothesis. *Annu. Rev. Neurosci.*, 18:555–86, 1995.
- [81] R.G. Smith. Neuronc: a computational language for investigating functional architecture of neural circuits. *J. Neurosci. Methods*, 43:83–108, 1992.
- [82] B. Stroustrup. *The C++ Programming Language*. Addison Wesley, 3rd edition, February 2000.
- [83] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, 1998.
- [84] K. Tanaka. Mechanisms of visual object recognition: monkey and human studies. *Curr. Opin. Neurobiol.*, 7:523–9, 1997.

- [85] A.M. Thomson and J. Deuchars. Temporal and spatial properties of local circuits in neocortex. *Trends Neurosci.*, 17:119–26, 1994.
- [86] S. Thorpe, D. Fize, and C. Marlot. Speed of processing in the human visual system. *Nature*, 381:520–2, 1996.
- [87] S.J. Thorpe and M. Imbert. Biological constraints on connectionist modeling. In R. Pfeifer, Z. Schreter, and F. Fogelman-Soulié, editors, *Connectionism in perspective.*, pages 63–92. Elsevier, New York, 1989.
- [88] E.C. Tolman. Cognitive maps in rats and men. *Psychological Review*, 55(4):189–208, 1948.
- [89] M.J. Tovee and E.T. Rolls. Information encoding in short firing rate epochs by single neurons in the primate temporal visual cortex. *Vis. Cogn.*, 2:25–58, 1995.
- [90] M.J. Tovee, E.T. Rolls, A. Treves, and R.P. Bellis. Information encoding and the responses of single neurons in the primate temporal visual cortex. *J. Neurophysiol.*, 70(2):640–54, 1993.
- [91] R. van Rullen, J. Gautrais, A. Delmore, and S. Thorpe. Face processing using one spike per neuron. *BioSystems*, 48(1-3):229–39, 1998.
- [92] P.F.M.J. Verschure and P. Althaus. A real-world rational agent: Unifying old and new ai. *Cognitive Science*, in press, 2003.
- [93] P.F.M.J. Verschure and P. König. On the role of biophysical properties of cortical neurons in binding and segmentation of visual scenes. *Neural Comput.*, 11:1113–38, 1999.
- [94] J.D. Victor and K.P. Purpura. Metric-space analysis of spike trains: theory, algorithms and applications. *Network: Comput. Neural Syst.*, 8:127–64, 1997.
- [95] C. von der Malsburg. Binding in models of perception and brain function. *Curr. Opin. Neurobiol.*, 5:520–26, 1995.
- [96] G. Wallis and E.T. Rolls. A model of invariant object recognition in the visual system. *Prog. Neurobiol.*, 51:167–94, 1997.

- [97] M.A. Wilson and B.L. McNaughton. Dynamics of the hippocampal ensemble code for space. *Science*, 261:1055–8, 1993.
- [98] L. Wiskott. Slow feature analysis: A theoretical analysis of optimal free responses. *Neural Computation*, 15(9):2147–77, September 2003.
- [99] R. Wyss, P. König, and P.F.M.J. Verschure. Involving the motor system in decision making. *Proc. R. Soc. Lond. B (Suppl.), Biology Letters*, in press.
- [100] R. Wyss, P. König, and P.F.M.J. Verschure. Invariant encoding of spatial stimulus topology in the temporal domain. *Neurocomputing 44-46*, pages 703–8, 2002.
- [101] R. Wyss, P. König, and P.F.M.J. Verschure. Invariant representations of visual patterns in a temporal population code. *Proc. Natl. Acad. Sci. USA*, 100(1):324–9, 2003.
- [102] R. Wyss and P.F.M.J. Verschure. Bounded invariance and the formation of place fields. In *Advances in Neural Information Processing Systems*, accepted for publication.
- [103] R. Wyss, P.F.M.J. Verschure, and P. König. Properties of a temporal population code. *Review in the Neurosciences*, 14(1-2):21–33, 2003.
- [104] K. Zhang, I. Ginzburg, B.L. McNaughton, and T.J. Sejnowski. Interpreting neuronal population activity by reconstruction: Unified framework with application in hippocampal place cells. *J Neurophysiol.*, 79(2):1017–44, 1998.

Curriculum Vitae

Name: Reto Wyss
Date and place of birth: 22nd of March 1975
in Scherzingen, Switzerland
Nationality: Swiss, citizen of Meggen (LU)
Marital Status: single

Education

- **2000-2003:** PhD at the Institute of Neuroinformatics, University/ETH Zürich
- **1997-2000:** Swiss Federal Institute of Technology Zürich (ETHZ), Dept. of Physics/Institute of Neuroinformatics
- **1995-1997:** Swiss Federal Institute of Technology Lausanne (EPFL), Dept. of Physics
- **1988-1992, 1993-1995:** Kantonsschule Alpenquai Luzern, Switzerland
- **1992-1993:** Hobart Senior High School, OK, USA (Exchange year)

Employment

- 1997-2003: teaching assistant for various courses at the ETHZ and the Institute of Neuroinformatics: Analysis III & IV for Electrical Engineers (1997-2000), Biophysics of Neural Computation - Introduction into Neuroinformatics (2001-2003), Physik I & II for Computer Scientists (2003).
- since November 2000: PhD student at the Institute of Neuroinformatics, University/ETH Zürich.
- 1999: practical work at the Institute of Neuroinformatics (work and thesis about a self-contained neural model for maze-learning).