Motion Perception and Prediction: A Subsymbolic Approach

Volker Baier

Institut für Informatik, Lehrstuhl für Theoretische Informatik und
Grundlagen der Künstliche Intelligenz

Motion Perception and Prediction: A Subsymbolic Approach

Volker Baier

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. A. Knoll

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Dr. h.c. mult. W. Brauer
2. Univ.-Prof. Dr. K. Schill
   Universität Bremen
3. Univ.-Prof. M. Beetz, PhD

Die Dissertation wurde am 28.06.2006 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 13.11.2006 angenommen.

Contents

List of Figures

List of Tables

Instead of the great number of precepts of which logic is composed, I believed that the four following would prove perfectly sufficient for me, provided I took the firm and unwavering resolution never in a single instance to fail in observing them.

The first was *never to accept anything for true which I did not clearly know to be* such; that is to say, carefully to avoid precipitancy and prejudice, and to comprise nothing more in my judgement than what was presented to my mind so clearly and distinctly as to exclude all ground of doubt.

The second, to *divide each of the difficulties under examination into as many parts as possible, and as might be necessary for its adequate solution.*

The third, to conduct my thoughts in such order that, by commencing with objects the simplest and easiest to know, *I might ascend by little and little, and, as it were, step by step, to the knowledge of the more complex*; assigning in thought a certain order even to those objects which in their own nature do not stand in a relation of antecedence and sequence.

And the last, in every case to make enumerations so complete, and reviews so general, that *I might be assured that nothing was omitted.*

René Descartes
Discourse on the Method of Rightly Conducting One's Reason and of Seeking Truth (Part II)

Abstract

The perception of dynamic spatio-temporal patterns, the perception of motion, is a fundamental part of visual cognition. In order to understand the principles behind these biological processes better, we analyze and construct a representation of dynamic spatio-temporal information on different levels of abstraction. Psychophysical experiments have shown that a spatio-temporal memory for early vision is evident and that the processing is realized in different representational layers of abstraction. The basic properties of this memory structure are reflected in multi-layered neural network models which this work is mainly about. Major architectural features of this network are derivatives of Kohonen's self-organizing maps (SOMs).

In order to render the system able to represent, process and predict spatio-temporal patterns on different levels of granularity and abstraction, the SOM's are organized in a hierarchical manner. The model has the advantage of a *self-teaching* learning algorithm and stores temporal information by local feedback in each computational layer. Prediction ability was integrated by the introduction of neural associative memories.

Constraints for the neural modeling and data sets for training the neural network are obtained by psychophysical experiments investigating human subjects' abilities for dealing with spatio-temporal information, and by neuro anatomical findings in the brain of mammals.

Zusammenfassung

Das Verarbeiten von dynamischen spatio-temporalen Mustern und die Perzeption von Bewegung, sind ein fundamentaler Bestandteil der visuellen Warnehmung. Um die Prinzipien auf denen Bewegungsperzeption beruht zu verstehen, und um die zugrundeliegenden biologischen Prozesse zu verstehen, haben wir die Repräsentation dynamischer spatio-temporaler Information in verschiedenen Verarbeitungsschritten analysiert. Psychophysische Experimente zeigten, dass ein spatio-temporales Gedächtnis in der frühen visuellen Verarbeitung evident ist, und dass die Verarbeitung in verschieden granularen Schichten erfolgt. Diese grundlegenden Eigenschaften eines visuellen Gedächtnismodells werden in dieser Arbeit in einem mehrschichtigen künstlichen neuronalen Netz realisiert. Dieses neuronale Netz stellt zudem das zentrale Thema dieser Dissertation dar. Die einzelnen Verarbeitungsschichten wurden von Kohonens selbstorganisierenden Karten abgeleitet und bilden den architekturelle Hauptbestandteil unseres Modells.

Wir haben eine hierarchische Strukturierung gewählt, um dem System die Repräsentation, Verarbeitung und der Prädiktion von spatio-temporalen Mustern auf multiplen Granularitätsebenen zu ermöglichen. Ein Vorteil hierarchischer Modelle ist die automatische Generierung der Eingabedaten für hierarchisch höhere, abstraktere Schichten. Die abstrakteren Schichten werden faktisch in einem generativen Prozess von den feiner granularen Stufen mit Trainingsinformation versorgt. Ein Kurzzeitgedächtnis wird in dem vorgestellten System durch lokale Rückkupplungen auf Neuronenebene realisert. Prädiktion wurde durch die Integration von neuronalen associativen Speichern ermöglicht.

Sowohl die Randbedingungen der Modellierung des neuronalen Netzes, als auch experimentelle Trainingsdatensätze, wurden durch psychophysische Experimente mit Versuchspersonen gewonnen und abgeleitet. Weitere Hinweise zur Modellierung wurden aus den Ergebnissen experimenteller Neuroanatomy gewonnen.

Acknowledgements

The research in this thesis was influenced by several research schools. I have to thank especially my supervisor Professor Dr. Dr. h.c. mult. Wilfried Brauer to guide me through my interdisciplinary project I started at the lab of Professor Dr. Kerstin Schill at the Institut for Medical Psychology. I am grateful to Professor Brauer for having the confidence completing my work at his chair, and for encouraging me to keep on going the path of neural network modeling. He supported me with an outstanding unconstrained research environment Professor Kerstin Schill, the second referee of my doctoral committee, encouraged and supported my theme already in the very beginning of my thesis. She and Professor Brauer gave me the opportunity to start my research within the graduate college "Motorische und Sensorische Interaktionen in biologischen und technischen Systemen". It was her research group which lead me, guided by many discussions, to my path to motion perception and prediction. Also the fruitful discussions with Christoph Zetzsche had important influence to my work. The third member of my doctoral committee was Professor Dr. Michael Beetz. I am very thankful for the great feedback on my work and for spontaneously agreeing to be my third referee. I would like to thank Professor Dr. Terry Caelli and Professor Dr. Walther Bischof for the great three month research stay I had at the University of Edmonton. The visit was supported by the Graduate College, Professor Brauer, Professor Schill, and Professor Caelli.

Matthias Kranz provided me with the gesture data set. This data set enabled me to conduct experiments close to real motion perception.

Not forgetting Felix Fischer, who helped proof reading this thesis.

Also many thanks to all my colleges at the IMP and chair VII, especially (in alphabetical order) Andreas Eisenkolb, Erol Osman, Florian Röhrbein, Elisabeth Umkehrer from the IMP and Felix Brandt, Markus Holzer, Michael Rovatsos, Gerhard Weiss, and Matthias Nickles from chair VII. Not to forget Professor Dr. Joseph Hochreither. He always is good address for fruitful controversy discussions. Last but not least I want to thank my parents supporting me in my study, and most important my wife Aneta, for being so patient with me during the final phase of my work. Moja draga, tvoja ljubav i podrška dala mi je snagu da završim ovaj rad. Volim te, Tvoj Volker.

Chapter 1

Introduction

Human beings have astonishing visual capabilities and do strongly depend on vision as their dominant sense of perception. As we are living in a highly dynamic environment, it is of crucial importance to process and predict motion of objects, including motion induced by ego-motion, fast and reliable. Motion processing is important for generating plans and the discovery of structures within trajectories. Motion prediction is important for aiming while hunting and more important in urban environments for obstacle avoidance.

To realize visual perception relevant tasks, the visual system of the human brain contains various processing layers, each specialized on a distinct task. An important field in research on the human visual system is the investigation of motion perception, motion prediction, and how motion processing is realized and organized in the cognition apparatus of the human brain.

In this dissertation we want to realize a motion processing and prediction system build of hierarchically structured artificial neural networks. To provide a good model it is important to based the models foundation on several scientific fields:

Cognitive psychology/psychphysics,
neuro-anatomy, and
machine learning, and artificial neural networks.

The goal we reach for is the definition and the construction of an artificial neural network model similar in its structure to biological brains visual motion processing capabilities, which is able to process and predict motion on several layers of abstraction. The question, how the human visual system is processing dynamic spatio-temporal motion patterns is approached from a multi-level perspective. Our model should realize information processing on various representational stages, as psychophysical experiments showed that the human visual system is performing its task in such a decompositional

fashion. Our investigations are focused on the architectural requirements of suitable memory structures for the representation of spatio-temporal information.

As a result, we introduce a basic spatio-temporal memory structure which provides necessary orthogonal access to spatial and temporal properties as already Schill and Zetzsche [SZ95] for a single processing layer. The processing of motion paths makes it necessary to introduce multiple hierarchically structured computing layers. Rendering the model capable of motion prediction and an autonomous completion of incomplete motion paths, it is necessary to introduce additional computational structures. We show several ways to realize the prediction capability.

The task of motion processing and prediction is quite difficult for a technical system, as it is mandatory to handle non-linearities within motion paths and to form compounds of motion for further –more abstract– processing layers autonomously. A technical system should be able –like the biological system– to generalize trajectories based on already observed data, and to predict and complete motion paths based on already learned motion paths. It should react robust on occlusions and all kinds of distortions.

A motion processing model should provide the ability to generate plans and detect structures within complex motion compounds autonomously. We further will refer to motion compound also as trajectories. Motion processing has to be organized in a hierarchical manner, as the cognitive processing layer is not able to process dense information. A hierarchical model is able to *compress* motion data, so the cognitive layer only has to handle extracted relevant information.

Motion prediction is important for aiming, targeting, pointing with a finger, hand eye coordination, and more important in urban environments, for obstacle avoidance. Catching a ball or a correct reaction to a Judo throw would be not possible without a prediction mechanism in the human visual system. The prediction system estimates the completion of the visual percept provided with only a small segment of the underlying trajectory, to enable humans to react just in time. A purely reflexive system would simply be to slow.

Our model is intended to generate autonomously representations of motion segments on different levels of granularities. It is designed in a multi-layered fashion, whereas the information propagation is delayed by predefined time interval. This allows the generation of compound information in hierarchically *higher* situated layers.

The key demands on our model are:

- Representation of motion information on several layers of abstraction, to allow an automatic integration of simple motion *atoms* to more complex motion compounds.

- The provision of a dynamic memory with orthogonal access to space and time within each single processing and the complete processing structure.
- An ability to process incomplete motion data and to automatically generate missing subsets of a perceived motion sequence.
- To base the whole system on research results gathered in biology, psychophysics, neuro-anatomy, ... as the human visual processing system is astonishing efficient, reliable, and fast.

Directly related to the question of how spatio-temporal information is accessed by the human visual system is the question of what kind of features are extracted and used by the visual system, and what kind of principal limitations do exist in dynamic information processing. Psychophysical experiments were carried out to identify these limitations. We use these limitations to constrain the artificial neural network model. One experiment for example addresses the discrimination capabilities of elementary motion patterns. A brief summary of the experiments conducted and latest results on these limitations will be presented in a separate section on the psychophysical background.

While being the main source of motivation, psychophysics is not the only foundation we use to develop our motion processing model. Pattern processing and prediction is a wide field providing several incarnations of artificial neural network models. Financial forecast applications for stock exchange markets for example are more and more dominated by artificial neural network systems. The models proposed in this work are able to represent information in several degrees of granularity. Therefore the models are able to generate a more abstract representation of the provided input data themselves. This automatic process can be interpreted as some clever kind of *low-pass filtering*.

Another foundation providing essential information is neuro-anatomy. As it is not possible to simulate artificial neural network assemblies built out of spike train neurons with a reasonable size, we had to balance the model between biological similarity and reasonable simulation time (which obviously includes the size of the neural assembly). Nonetheless we want to shift the design of our model to a biological more plausible realization, therefore we investigated improvements of the design to established models and novel models based on Kohonens self organizing maps.

Our cognitivistic motion processing and prediction model reflects elements of inspiration and architectural design hints from philosophy, cybernetics, psychology, neuro-anatomy, cognitivism, behaviorism, and connectionism.

To realize the argued demands we developed several neural network structures, implemented these networks in matlab, and conducted some experiments.

- We introduce an orthogonal access memory enabled self-organizing maps based algorithm, to allow the system to store context information already within the first processing structure in temporally ordered fashion.
- A learning mechanism which allows the network to unfold in the beginning of the training phase, to prevent twists in the network topology, and to conserve the temporal structure of the input data.
- A prediction mechanism based on a neural hetero associative memory simulating an indexing structure for retroinjecting lines to lower the threshold of the next most likely to be activated neuron.
- We introduce several multi-layer models capable of the key demands.
- To benchmark our system, we develop two single layer structures for sequence processing and automatic sequence reproduction.
- The implementation is realized in matlab, to gather empirical data on the behavior of our newly defined networks. This implementation is based on the somtoolbox for matlab.

Organization of the Thesis

We divide this work into several streams discussing motion processing from various points of view, finally ending up in the central topic: Motion perception and prediction realized with a subsymbolic approach.

- Chapter 2: In the next chapter we will give a short introduction and an overview on our motion processing and prediction model.
- Chapter 3: Introduces some paradigms of perception, to give an idea of the roots of the course of perception research.
- Chapter 4: As we want our model to foot on findings in real biology, it is necessary to give a small introduction to the visual system in real biological brains.
- Chapter 5: In our case, biology has not been the main source of motivation for our motion perception model, as you might notice later. The functionality of the system is based on psychophysical experiments and the derived results from the experiments were gathered at the Institute

of Medical Psychology LMU Munich. The experiments and the surrounding projects were funded by the DFG priority program "Spatial Cognition".

- Chapter 6: Processing dynamic data with artificial neural networks needs some static neural network basics. These are discussed in chapter 6.

- Chapter 7: This chapter gives an introduction to various realizations of short-term memories which may be added to static neural networks.

- Chapter 8: The next natural step following the last two chapters is the combination of short-term memory and static neural networks to form artificial neural networks capable of spatio-temporal information processing. Chapter 8 introduces novel approaches using single-layer models.

- Chapter 9: Provided with all these basics, we are now able to combine single processing layers to multi-layer neural networks capable of processing and predicting spatio-temporal data. First we again discuss established models to point out the improvement we realized in our novel approach.

- Chapter 10: As we have the definition of an artificial neural network model capable of all our requirements, we can proceed with some experiments.

- Chapter 11: Some conclusions and experimental results arising from our work, which are discussed in this chapter.

- Chapter 12: Finally, research in the area of motion processing does not end here. This work of course gives us some ideas what direction to go next.

Chapter 2

Overview of the Motion Processing and Prediction Model

The goal of this dissertation is as already mentioned the development of a multi-layered self-organizing maps based model that is capable of motion processing and motion prediction. The models realization follows ideas we derived from several research schools. To meet the demands of these schools we will introduce several extension to self-organizing maps as well as novel multi-layer structures and prediction mechanisms.

The sketch in Figure 2.1 shows a simplification of the final model introduced in detail later. To point out the improvements and the novelties in our model we will now give a brief introduction to it.

Before we are able to define a perception model we have to define the input of the system first. As input we argue for a sequence of spatio-temporal orientations which is a temporal sequence of vectors representing the two qualities of orientation and velocity. The stimulus might be interpreted as a single black dot moving on a white board, or more general a non relevant or specific background. Other visual qualities like shape and color or textures are assumed to be already processed by different computational mechanisms similar to the processing in the brain.

This input is represented to a self-organizing map based processing layer, which we extended to be able to store information on neural activation events in a temporally ordered fashion, in an orthogonal access memory. This conserves context information for several time steps, controlled by a damping factor.

The temporal storage can be interpreted as a short-term memory. Every single STORM layer has the capability of a simultaneous orthogonal access to the spatial position of the stimulus and the temporal occurrence of it.

As we want to process temporal activation patterns in hierarchically

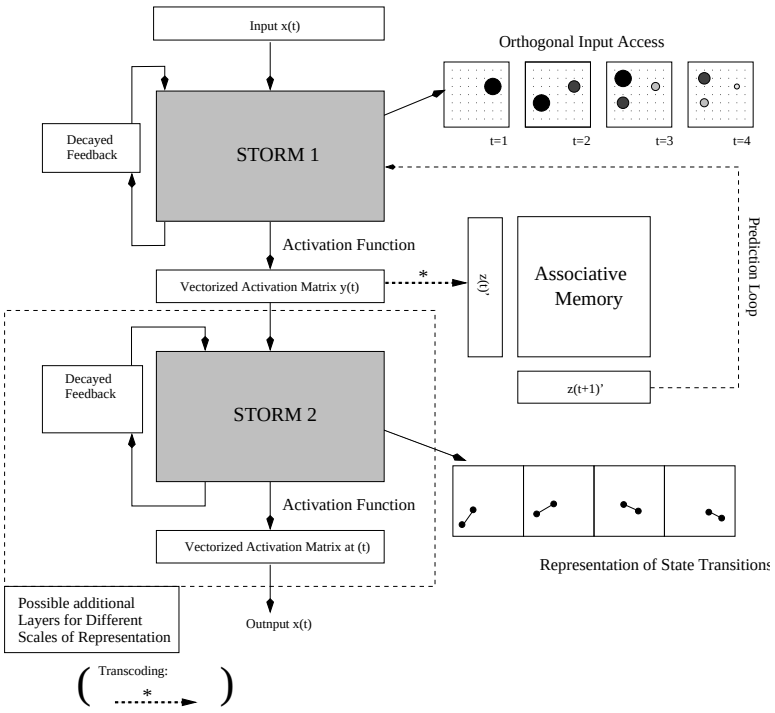

Figure 2.1: Spatio-Temporal STORM. The first layer processes a spatio-temporal input to an activation mapping which is propagated to the next higher processing level after several time steps. The second STORM layer processes compounds of state transitions of the first layer.

higher, more abstract structures we developed an extension to recurrent self-organizing maps, which ensures the temporal ordering.

In our new STORM layer model which we developed, a currently activated neuron is set to be refractory. The neuron is taken out of the pool of competing neurons the next few time steps. The time interval is indirectly controlled by a damping factor. After the refractory period is over, the neuron is included into the pool of competing neurons again. This processing scheme is resulting in an ensured fixed number of *active* neurons.

The information we want to transport from one hierarchy level to the next is composed by a textual concatenation of all activation values from one computing layer. This vector is composed after a predefined time interval based on psychophysical experimental findings –finger tapping experiment, flicker fusion–. This scheme is resulting in an activation vector, not as sparse as in the established models which often suffer the disadvantage of nearly useless intermediate processing layers.

Using our STORM model it is also possible to calculate the number of maximum possible patterns a layer is able to generate. Given the number

of maximum patterns it is possible to taylor the size and shape of higher processing layers.

We developed a training mechanism for temporal data which preserves the temporal context of the training data without having the disadvantage of badly adapted network structures.

The demand of motion prediction capability is met by the introduction of binary hetero-associative memories as prediction memories. This well known model acts as an indexing structure to lower the activation threshold of the next most likely to be activated neuron. As we use a quite different paradigm to temporal forecasting we cannot use the prediction mechanisms of the established recurrent self-organizing maps based models, so we add associative memories to learn the association of current activation status and most likely activation status in the next time step.

Psychophysics and neuro-anatomy provide us with hints on several design issues and parameter choice. The hierarchical structuring, the temporal limitations of the short-term memory, and the organization of the associative memories were guided by these disciplines. Our model is able to provide an orthogonal access to past activation events without any additional computational effort to allow context aware processing, demanded also by the dynamic spatio-temporal memory model proposed by K. Schill and Ch. Zetzsche [SZ95].

The associative memories provide prediction capability and the output of the associative predictor inherently provides a rating of the generated output. The hierarchical structuring allows us to operate on several scales of abstraction. Provided with enough processing layers, the model is able to classify simple motion plans by composing prototypical motion events to motion compounds.

Our approach goes beyond the orientation selective filters and Reichardt detectors, to the next *higher* processing abilities of the human visual system, it is a model, describing and simulating an integral part of the human visual system and it might be the last preprocessing structure before entering cognition.

Chapter 3

Paradigms of Perception

We only look, but we don't see!

Andrej Tarkowskij (russian film director, 1932-1986)

The *act* of perception, apperception and thinking is investigated in numerous disciplines. In this chapter we will introduce those schools that influenced and inspired our model finding process. We will start with the first discipline at all to deal with the act of thinking, perception, and cognition: philosophy.

3.1 Philosophy

Philosophers were the first scientists who tried to explain active cognition, thinking and self-awareness in a cognitive theory. Discussing all philosophical cognition theories would of course go beyond the scope of this work. We therefore constrain the discussion on two exemplary philosophers whose work directly influenced our development of a motion processing model.

Gottfried Wilhelm Leibniz introduced in his book Monadologie (1714) the term of apperception as a notation for the active self-awareness, what he also called the *monad*. The *monad* fulfills the transition of the passive perception process to active reflected perception.
Immanuel Kant refined this concept by subdividing it into empirical and transcendental apperception. His work *Critique of Pure Reason* (1781) is introducing the *Transcendental Esthetic* as the foundation of perception. Kant states that we have an external sense which gives us an idea of space and we also have an internal sense which gives us the idea of time. Space and time are both necessary for enlightenment. We are not able to imagine artifacts without space and time. At the same time our senses are receptive. *Kants Kopernikal turn* states that we do not recognize a thing itself but rather its emergence. This emergence is formed and generated by ourselves

through our intellect, by a kind of generative process.
However Leibniz and Kant were not the first philosophers who thought about how the human mind might work, but the cited ideas show quite obvious links to some schools of artificial intelligence, connectionism, and cognitive modeling. We partially base this thesis on reactions to phenomena observed on testing subjects doing psychophysical experiments. The hierarchically higher processing layers of our model are provided with an input resulting out of a generative process of the lower processing layers.

We now continue with a few sections on established paradigms of perception models which will support us on finding a reasonable motion perception model.

3.2 Behaviorism

Behaviorism has been a reaction to the previous use of introspection as a methodology of self analysis. On a closer look behaviorism is quite similar to the act of perception as described in Kants work, the differences lies in the absence of introspection.

The two main criteria of behaviorism are first, that any data should be publicly observable thus, as mentioned, excluding introspection which is discarded completely and second, that only the *behavior* of animals and humans should be studied avoiding mentalistic topics such as thinking, imagination, intentions, desires, plans, symbols, schemes, or other mental representations completely.

In the eyes of behaviorists all psychological activity can be explained without the need to resort to such mentalistic entities. The most important factor in generating the behavior of individuals is the environment. Individuals are seen as passive reflectors of various forces and factors in their environment.

General principles and methods of conditioning and reinforcement where developed by behaviorists to explain how learning and shaping of particular behaviors could be established. Behaviorists state that the science of behavior could account for a complete description of everything an individual might be able to do, explicitly including the act of thinking which is assumed to be a covert behavior.

The paradigm of behaviorism influenced research until the mid twentieth century. Difficulties associated to introspectism were overcome by simply discarding vague and weakly defined concepts like *will* or *purpose*. Behaviorism oriented itself on extrapolating observations of animal behavior to human behavior. Think of Pavlovs experiments on conditioning a dog to an acoustic stimulus to produce more salvia as a simple example. This experiment was also run with human testing subjects with similar results. But as many behaviorists ignored all discussion of important topics like language,

problem solving, planning and imagination –which in fact is quite similar to introspection–, they provided a huge point of attack for other research schools. Most behaviorists not only rejected to deal with the aspects of cognition they even did not tolerate cognition. Skinner, representing a quite radical behaviorism on the other hand, allowed for cognitive aspects, in contrast to most other researchers. Skinners *Verbal Behavior* is an analysis of speech in terms of *controlling relations* which include the speaker's current motivational state, his current stimulus circumstances, his past reinforcements, and his genetic constitution. Pure behaviorism caused an opposition to the perceived sterility of behaviorism. Skinners book *Verbal Behavior* was released in 1957 [Ski57]. Two years later Naom Chomsky delivered a review on it which went into history under the designation *Demolition of Behaviorism*. This rather destructive criticism lead nearly to a total discard of behaviorism this stasis lasted at least for ten years and still resides in the mind of many researchers until now, whereas behaviorism contributed many important research results like the principle of reinforcement learning.

Also in the 50's and 60's a debate arose for the need of a theory of human cognition to account for complex organized behaviors such as the use of language, playing strategic games which involve planning, or playing an instrument. Lashley [Las51] argued for a serially ordered chain of behaviors and dropped the paradigm of strict associative stimulus response chains. One argument was that for example, when playing a musical instrument the time constraints "left no time for feedback, no time for the next tone to depend upon or in any way to reflect the course of the previous one". Further, slips of the tongue often anticipate words that only occur much later in a sequence. These events, Lashley claimed, cannot possibly be explained by simple linear *A evokes B* event chains. They must be planned and organized in advance, what in fact represents an early demand on a prediction capability of the processing system.

3.3 Cognitivism

In the late 1940s, with the invention of the first computers, numerous researchers were beginning to realize that a kind of similarity between the way a brain works and the way computers work exists. The Hixon Symposium held in 1948 can be seen as the starting point of modern cognitive science. It was originally designed to discuss the way the nervous system controls behavior. John von Neumann held the opening talk within he made a comparison between the human brain and electronic computers [Gar85].

To characterize cognitivism we can state that coginitivism consists of two major components, one is of methodological nature and the other one of theoretical nature.

Methodologically, cognitivism adopts a positivist approach and the belief

that psychology can be –in principle– fully explained by the use of experiments, measurements and the scientific method. This is also largely a reductionist goal, with the belief that individual components of mental function the –cognitive architecture– can be identified and *meaningfully* understood. The second is the belief that cognition consists of discrete, internal mental state representations or symbols whose manipulation can be described in terms of rules or algorithms.

Also in the late 1940s an alternate paradigm for cognition attracted support: connectionism. Cognitivism and connectionism found wide support in parallel until the 1960s, when due to a book of Minsky and Papert connectionism suffered a serious setback [MP69]. The setback lasted for about twenty years, unless some new learning algorithms and network models were introduced. Building computer models and running those models became the frontmost paradigm and is still the major research program in Cognitive Science, Artificial Intelligence (including Computer Vision and Machine Learning), Cognitive Psychology and Linguistics. Computer models are denominated in literature with various names –cognitivism, symbol processing system and representationalism. We will generally stick to the term cognitivism. Cognitivism was also born as a reaction against pure behaviorism. Where behaviorism excludes the processes of the 'mind', cognitivism concentrates especially on isolated mental processes.

Dreyfus & Dreyfus [DD86] stated that cognitivists supported a belief that the human brain and digital computers have a common functional description, on a certain level of abstraction. They argue for the similarity of the computational power of both systems which both can generate intelligent behavior through symbol manipulation being programmed in a proper way.

> "... the human brain and the digital computer, while totally different in structure and mechanism, *have* at a certain level of abstraction a common functional description. At this level both the human brain and the appropriately programmed digital computer *can* be seen as two different instantiations of a single species of device–a device that *generates* intelligent behavior by manipulating symbols by means of formal rules."

This view shows that both, mind and digital computers, can be interpreted as symbol processing systems, at least in the framework of Dreyfus & Dreyfus, which also shows the interrelationship to deduction mechanisms and logics in philosophy. Mind and digital computers can nonetheless not be seen as equivalent systems. The physical symbol system hypothesis discussed by Dreyfus & Dreyfus was originally promoted by Newell & Simon [NS81]. Newell & Simon stated that:

> "A physical symbols system has the necessary and sufficient means for general intelligent action. ... By 'necessary' we mean that any

> system that exhibits general intelligence will prove upon analysis to be a physical symbol system. By 'sufficient' we mean that any physical symbols system of sufficient size can be organized further to exhibit general intelligence."

Cognitivism demands a formalized representation of the real world, and looks to logic to provide the linking mechanism. This view follows a long history of rationalist, reductionist tradition in philosophy, going back to at least the time of Plato. This tradition was enhanced by Descartes, who made the assumption that all understanding required the creation and manipulation of appropriate representations, and that these representations could be analyzed into primitive elements –naturas simplices– such that all phenomena could be understood as complex combinations of these simple elements.

Other philosophers who made contributions include Hobbes –who made the suggestion that the elements were formal components related by purely syntactic operations, so that reasoning could be reduced to calculation–, Leibniz –who posited that every human concept must be composed of complex combinations of *ultimate simples*, and if these concepts are to apply to the world, there must be simple features that these elements represent–, up to Frege and Russell. Wittgenstein contributed his apotheosis of the reductionism, rationalist tradition (*Tractatus Logico-Philosophicus*), which postulated a pure form of this syntactic, representational view of the relationship of the mind and a world composed of facts.

Artificial Intelligence is the continuation of this philosophical tradition. AI attempts to determine primitive elements and logical relationships that mirror the primitive objects and their relationships.

3.4 Cybernetics and Constructivism

The school of cybernetics was founded independently by several researchers. Some of them origin from physics, some of communication engineering, maths, psychology and human biology. Some of the most important researchers are Warren McCulloch, Norbert Wiener, Frank Rosenblatt, Heinz von Förster to name a few.

Cybernetics to characterize the term, is the study of communication and control involving regulatory feedback, in living organisms, in machines, and in combinations of the two, for example, in sociotechnical systems. Louis Couffignal characterized cybernetics as:

> "Cybernetics is the art of ensuring the efficiency of action"

The epistemology of cybernetics is constructivism. Ernst von Glasersfeld is one of the founders of the radical constructivism. He defined the radical constructivism by the following two basic principles:

- Knowledge is not passively received either through the senses or by way of communication, but is actively built up by the cognising subject.

- The function of cognition is adaptive and serves the subject's organization of the experiential world, not the discovery of an objective ontological reality.

In principle radical constructivism states that every single human is constructing his own reality by interpreting his perceptions. So every human is generating his own world within he is interacting and perceiving. All processes for perception are invented independently within every living entity. In our opinion this point of view is to strict, since neural anatomy has provided us with findings which are not bound to an a single human more-as found in each examined testing subject. It is in our opinion evident that every reflected entity is constructing its own belief of a world, but provided stimuli result in similar excitation patterns in every human brain.

Researchers from the biological cybernetics school try to find an information theoretical formulation to a whole organisms, whereas neuro cybernetics tries to find proper communication engineering based description of neurological models to explain perception phenomena.

3.5 Connectionism

Following a serious setback in the 1960s, caused partially by a massive critique of the power of perceptrons by Minsky & Papert [MP69], the connectionist approach had since the early '80s a resurgence in popularity. The awakening in the early '80s was partially caused by the book series "Parallel Distributed Processing" by Rumelhart and McClelland [RM86].

Current research in connectionism concerns feed-forward style networks (Rumelhart, Hinton & Williams '86, Stone '86, ...) Kohonens self-organizing maps [Koh97], various recurrent networks, adaptive resonance theory (ART) to name some but not all of the known models. Review and general references can be found in books on artificial neural networks e.g. Haykin [Hay94]. Some of the researchers dealing mainly with pattern processing and pattern recognition turned in the last years to purely statistical models like support vector machines (SVM), principal component analysis, hidden markov models, dynamic bayesian networks or in the case of dynamic data processing to kalman filtering or particle filtering. The turn of favor to statistical methods was also caused by the fact, that neural network models are not completely observable like in contrast SVMs, but which in fact only is more of an artificial advantage.

Nonetheless our belief is that a cognition system capable of motion perception should be as biological plausible as possible. So our choice is to stick

to self-organizing maps. In our opinion biological systems rely on two learning mechanisms: evolutionary tuning of the population shape and spatial structure, number of neurons and interconnection fibers and on synaptical weight adaptation. This forces one almost to favor models like self-organizing maps, where the choice of the neighborhood function, the mesh size can be associated to evolutionary learning, the adaptation of the neuron mesh-grid could be referred as the maturating part of the adaptation of the system, and the pure weight adaptation which could be interpreted as the synaptical adaptation.

The most common learning rule which is used within feed-forward models, error back-propagation, is of a questionable biological validity. Back-propagation learning is based on the determination of an error difference between the current output vector of the network and its expected output vector. In biology we normally will not be provided with an expected output, so we will not be provided with a final result of a task as the environment is highly dynamic. We belief in the existence of self-organizing processes, which allow the human brain to adapt itself to whatever statistics it is confronted with.

A general theory of cognition should not 'design a net' as in some localist networks, but rather have a general structure that is able to learn for itself –otherwise the net will only exhibit the expert knowledge built into it by its designer. Intelligent acting is determined by motivations and purposes within an organism, as well as successful behaviors picked up by the organism from an ongoing external world and culture.

3.6 Computational Neuroscience

Computational Neuroscience is an interdisciplinary school of research. Fundamental works were published by Huxley, Hodkin, and Marr. Huxley and Hodkin developed the voltage clamp which allowed the development of the very first model of the action potential. David Marr concentrated on interactions between neurons and suggested computational approaches on how functional assemblies of neurons within hippocampus and neocortex interact with each other, store and process and transmit information. Computational Neuroscience is the attempt to simulate neurons on a level which is as exact as possible bound to biology, to understand the interrelationships between neurons and synapses and to provide a theoretical framework for neural processes. Especially the coding of information with spikes and the meaning of syncronicity in spike patterns are a point of interest. A possible statistical description of spikes can be realized with Fokker Planck equation based models. We took computational neuroscience rather than a source of inspiration how to trim our model further to a biologically realistic model. At the current state of research it is not possible to simulate big popula-

tions of neurons. Nicolas Brunel did simulations on up to a maximum of ten neurons recently (2004). As computing power rises and better approximations on spike train models arise, computational neuroscience gets more and more interesting, but for now we have to find a path somewhere in the middle of connectionism and computational neuroscience ever shifting the focus more and more to the latter school. So possibly the developments of neural computation and our attempts might converge in between the two schools in near future. Nonetheless it should be stated that the functions used in connectionist neural networks are the result of summing up action potentials over time. So connectionist networks also base on spiking neurons, only on a more coarse scale of time.

3.7 Guiding Concepts

Finally we want to summarize some ideas we want to transfer to our motion perception model. Behaviorism and cognitivism are based on philosophical discourses of perception we described at the very beginning of this chapter. Cognitivism deliver the functional background and most of the constraints applied to our system. As our model will be built from artificial neural networks based on self-organizing maps propagating information from one processing layer to the other, it is not fully introspectible, what makes it similar to a behavioristic approach or speaking in the sense of Heinz von Förster a *non trivial machine*. As we use self-organzing maps based layers and neural associative memories as we will notice later, we define a kind of connecitionist model. Neural Computation is as mentioned more of a distant goal to reach, as we try to refine our model to fit better to real biological findings. Also the size of simulate-able neuron population is simply to small for our purpose. Overall we can call our attempt of *inventing* a motion perception model a constructivistic approach. Cybernetics is present by the feedback in each computational entity of our model. We will deliver a full argumentation line in the conclusion as we have every part of the model then.

We will now continue with some neuro-anatomical findings of the visual system to find some structural hints on how an artificial motion processing and prediction model could look like.

Chapter 4

Neuro-Anatomy of Visual Perception

There are two ways to get an idea how motion perception might be realized in the brain. One way is to present simple motion stimuli in experimental conditions to –non human– testing subjects while doing single cell recordings with platinum electrodes or fMRI to see which neuron patches respond to the presented stimuli. The other way, discussed in the following chapter, is to present testing subjects specially designed motion stimuli and measure their reaction time or accuracy to these presented stimuli. By interpreting the reactions it is possible to develop models. These kind of models describe the functionality of the system of course more, as they are anatomically reasonable.

This chapter will give a short introduction to the neuro-anatomy of the visual system, as we have also taken it into account to develop our motion perception model to get a biological plausible structure.

4.1 Visual Processing Stream in Cortical Areas

While cognition psychology gives us an idea of the functional entities of the whole system, neuro-anatomy provides us with information on how areas and neurons are interconnected with each other to form the visual processing system. The stream of visual processing is divided into two separate pathways with different functionality. We will mainly concentrate on the dorsal pathway of the visual processing system, since motion perception is realized on this processing stream. The main focus lies on the areas involved in motion processing. A quite comprehensive description of the visual system can be found in [RD02].

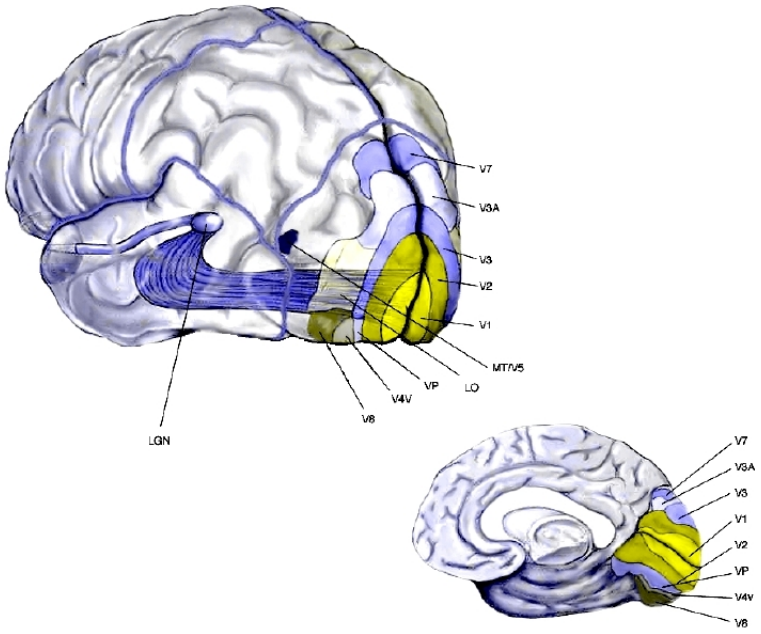

Figure 4.1: Sketch of the location of the visual perception related structures in the human brain.

4.1.1 The Dorsal Pathway

In the cortex [And97] motion is processed in a hierarchical fashion as illustrated in Figure 4.2. The first processing stage in the hierarchy consists of simple and complex cells that are direction selective and situated in the cortical visual area V1. Evidence suggests that directional selectivity in the primary visual cortex is computed independently from retinal direction selectivity. The direction selective cells in V1 project to the middle temporal cortical area (MT or V5) and to V2, which also projects to MT. Directional selectivity becomes more complex in MT; that is, cells there typically have very large, orientation-independent receptive fields, and many will respond best to the composite motion of a plaid, as opposed to its individual components. From MT, the motion pathway projects to MST, wherein directional selectivity information is further combined to produce neurons sensitive to complex motions, such as rotation, expansion and contraction [And97]. Fundamentals can be found in two outstanding works of S. M. Zeki: [Zek74], [Zek78]. The structuring of the dorsal processing stream provides us with some constraints of the hierarchical structuring of our artificial neural network model.

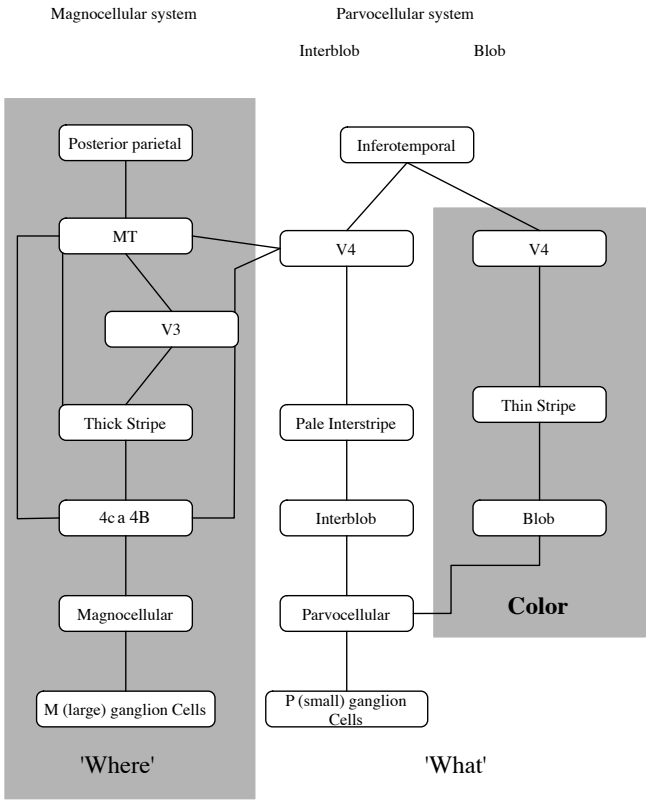

Figure 4.2: Simplified overview of the visual pathway.

4.1.2 The Ventral Pathway

The ventral stream begins with visual area V1, goes through visual area V2, then through visual area V4, and to the inferior temporal lobe. The ventral path, also called the "*What Pathway*", is associated with shape recognition and object representation. It is also associated with the storage of long-term memory representations. We can say that the ventral stream is only related to static pattern processing unlike the dorsal stream, which processes also dynamic patterns.

4.2 Visual Cortical Areas

We want to discuss the visual cortical areas in a very brief way, to give an idea how visual processing is realized in real brains and to gather some more constraints for our network model.

4.2.1 Retina and Lateral Geniculate Nucleus

The retina converts and filters the incoming light stimuli into electrical action potentials. It is composed out of approximately 6 million cones and 100 million rods, whereas cones are tuned for the perception of single colors –red, green, blue– and fast movements but less light-sensitive than cones and rods are more light-sensitive and responsible for the perception of luminance –black, white– changes. Cone photo-receptors also show a temporal property. They behave like temporal bandpass filters. The peak response of these filters is reached at a frequency of 5Hz [Bay87], the so called flicker fusion frequency.

The flicker fusion frequency also defines the frequency when the strongest motion aftereffects occur [Pan74]. The existence of the flicker fusion frequency is useful to know, because it allows us to constrain phsychophysical experiments and it delivers a good hint for parameter setting and finding in artificial neural network models.

Retinal ganglion cells and lateral geniculatie nucleus (LGN) neurons can be subdivided into sub-populations with different spatial and temporal properties [Len80].

The LGN is a sensory nucleus situated in the thalamus of the brain which receives its inputs from the retina and projects them retinotopicaly to area V1. The LGN anatomical structure is subdivided into six functional layers. Layers 1, 4 and 6 correspond to signals from the one eye; layers 2, 3 and 5 correspond to signals originating from the other eye. Layer 1 consists of M cells, which correspond to the magnocellular (M) cells of the optic nerve of the opposite eye, and are involved in depth or motion processing. Layers 4 and 6 of the LGN also connect to the opposite eye, but to the parvocellular (P) cells of the optic nerve. Those P cells are sensitive to color and edges. Layers 2, 3 and 5 of the LGN connect to the M and P cells of the optic nerve corresponding the same side of the brain as represented in LGN. In between the six layers are smaller cells receiving signals from the cones in the retina. The neurons of the LGN then relay the visual stimuli to the primary visual cortex V1.

4.2.2 Primary Visual Cortex V1

The primary visual cortex is the best studied visual area of the brain so far. It is the part of the cerebral cortex that is responsible for processing visual stimuli. It is the simplest, earliest cortical visual area. It is highly specialized for processing information about static and moving objects and is excellent in pattern recognition.

The functionally defined primary visual cortex is approximately equivalent to the anatomically defined *striate cortex*.

V1 has a well-defined map of spatial information. For example, in the

human brain the upper bank of the calcarine sulcus responds strongly to the lower half of visual field (below the center), and the lower bank of the calcarine to the upper half of visual field. Conceptually, this so called retinotopic mapping is a signal transformation of the visual image from the retina to V1. The correspondence between a given location in V1 and in the subjective visual field is very accurate: even the blind spots are mapped into V1. Evolutionary, this correspondence is very basic and found in most animals that possess an area similar to V1. In human and animals with a fovea in the retina, a large portion of V1 is mapped to the small, central portion of visual field, a phenomenon known as cortical magnification. A technical representation of this foveal coritcal magnification can be realized with a log-polar mapping of images. Perhaps for the purpose of accurate spatial encoding, neurons in V1 have the smallest receptive field size of any visual cortex regions.

Individual V1 neurons have strong tuning to a small subset of stimuli characteristics. They respond to small changes in visual orientations, spatial frequencies and colors. Furthermore, individual V1 neurons in humans and animals with binocular vision show an ocular dominance, they are responsible for the tuning to one of the two eyes. In V1, and primary sensory cortex in general, neurons with similar tuning properties are clustered together to so called cortical columns shown in Figure 4.3. D. H. Hubel and T. N. Wiesel [HW77] proposed the ice-cube organization model of cortical columns for two tuning properties: ocular dominance and orientation. However, this model cannot accommodate color, spatial frequency and many other features to which neurons are tuned. The exact organization of all these cortical columns within V1 still remains an important topic of current research. It seems to be widely accepted that area V1 consists of tiled sets

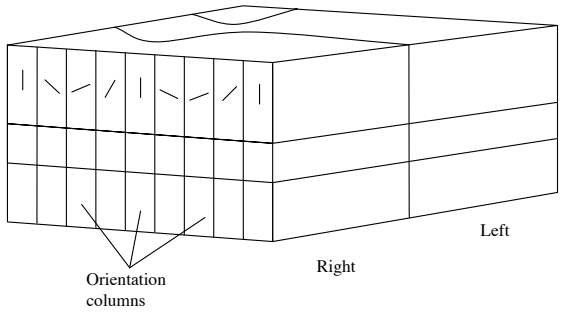

Figure 4.3: Ice-cube organization of orientation selective cells in V1

of selective spatio-temporal filters. In the spatial domain, the functioning of V1 can be thought of as similar to many spatially local, complex fourier transforms. Theoretically, these filters can carry out neuronal processing

of spatial frequency, orientation, motion, direction, speed (thus temporal frequency), and many other spatio-temporal features.

The visual information propagated to V1 is not coded in terms of spatial (or optical) imagery, but rather as the local contrast. As an example, for an image comprising half side black and half side white, the dividing line between black and white provides the strongest local contrast and is encoded, while few neurons code the brightness information. As information is further relayed to subsequent visual areas, it is coded as increasingly non-local frequency/phase signals. Importantly, at these early stages of cortical visual processing, spatial location of visual information is well preserved amid the local contrast encoding.

Visual Area V2

Visual area V2 first region within the visual association area. It receives strong feed-forward connections from V1 and sends strong connections to V3, V4, and V5. It also sends strong feedback connections to area V1.

Anatomically, V2 is split into four quadrants, a dorsal and ventral representation in the left and the right hemispheres. Together these four regions provide a complete map of the visual world.

Functionally, V2 has many properties in common with V1. Cells are tuned to simple properties such as orientation, spatial frequency, and color. The responses of many V2 neurons are also modulated by more complex properties, such as the orientation of illusory contours and whether the stimulus is part of the figure or the ground [QdH05].

Recent research has shown that V2 cells show a small amount of attentional modulation (more than V1, less than V4), are tuned for moderately complex patterns, and may be driven by multiple orientations at different subregions within a single receptive field.

Visual Area V3

Visual area V3 is part of the dorsal pathway, receiving inputs from V2 and primary visual areas. V3 projects to the posterior parietal cortex.

Recent research based on fMRI recordings suggested that area V3/V3A could play a role in the processing of global motion [BQ01].

Visual Area V4

V4 is the third cortical area in the ventral stream, receiving strong feed-forward input from V2 and sending strong connections to the posterior inferotemporal cortex (PIT). It also receives direct inputs from V1, especially for central space. It has also weaker connections to V5 and to visual area DP (the dorsal prelungate gyrus). V4 is the first area in the ventral stream to show strong attentional modulation. A paper by Moran and Desimone

[MD85] characterizing these effects was the first paper to find attention effects anywhere in the visual cortex [Zek74],[Zek78].
Like V1, V4 is tuned for orientation, spatial frequency, and color.
Unlike V1, it is tuned for object features of intermediate complexity, like simple geometric shapes, although no one has developed a full parametric description of the tuning space for V4. Visual area V4 is not processing complex objects such as faces. Faces are processed in the inferotemporal cortex. Zeki argued that the purpose of V4 was to process color information. Work in the early 1980s proved that V4 was as directly involved in shape recognition as earlier cortical areas. This research supported the Two Streams hypothesis, first presented by Ungerleider and Mishkin [UM82]. Recent work has shown that V4 exhibits long-term plasticity, encodes stimulus salience, is gated by signals coming from the frontal eye fields, shows changes in the spatial profile of its receptive fields with attention, and encodes hazard functions.

Visual Area V5

Visual area V5 also known as area MT, is a region that appears to process complex visual motion stimuli. It contains many neurons which are selective to the motion of complex visual features like end-stop- and corner sensitivity. Much work has been carried out on this region as it appears to integrate local visual motion signals into the global motion of complex objects [MAGN85]. There is still much controversy on the exact computations carried out in area MT [WR92]. Some research suggests that the feature *motion* is in fact already available at lower levels of the visual system such as V1. Bullier [Bul01] investigated retroinjecting lines from higher visual processing areas including V5 to V1, providing an active blackboard which supports motion prediction. One observable phenomena is apparent motion.

Visual Area V6

This functional area was so far only identified in monkeys and not in human brain.

Visual Area V7

The function of area V7 is so far unknown.

Visual Area V8

Area V8 in the human brain is processing color information and is believed to be the homologous to macaque brain area V4. It was investigated by using fMRI by Hadjikhani et. al. in 1998 [HLD$^+$98].

4.3 Motion Processing Related Areas

Motion processing is as the last subsections showed not located in a singular area. The processing is distributed along a pathway including the retina, LGN, V1 projecting to MT and V2, V2 projecting to V3/V3A and MT which projects to MST and MST to area 7A. We illustrate this pathway in Figure 4.4 by indicating the areas involved in motion processing with a grey shading. This sketch is of course not complete it only shows the most important functional areas.

The Role of Retroinjecting Connections

The main limitation of feed-forward models is that they do not combine detailed local analysis with a global percept, necessary to resolve occlusion and lightning artifacts. It is therefore necessary to use recurrent and retroinjecting communication lines [Bul01].

To compute interactions between distant regions in the visual field, two types of connections are used in the brain. Local horizontal connections and feedback connections. The necessity of computing with high spatial precision and reaching out to distant regions in the visual field is difficult to achieve only with horizontal connections within a single cortical area. With local horizontal connections, each V1 and V2 neuron is limited to computations in a very local environment and it cannot participate in integration across long distances in the visual field. Neurons in higher order areas like MT, V4, TEO, are able to integrate information across long distances in the visual field because of their larger receptive fields and the lower magnification factors in these areas. However, the selectivity of their receptive fields are more specialized. For neurons in higher order areas, long-distance integration can be properly achieved through horizontal connections for a given aspect of computation corresponding to the major selectivity of the area like the computing of the direction of movement in area MT. Nonetheless, such computation is impossible when it requires combining neurons with selectivity to different attributes like motion direction, depth, color, shape etc. . . simultaneously.

One possible way to achieve this combination is by exchanging information between neurons in higher order areas coding for different attributes. However, the level of complexity of the computation and the fine grain of the representation that is often needed are probably impossible to achieve with the rather sparse set of such connections like for example the small number of direct connections between higher order areas of the dorsal and ventral stream.

Computation across long distance in the visual field is realized by retroinjecting the results of the computations done by neurons in higher order areas

through feedback connections to neurons in low order areas such as V1 and V2. Contrary to local horizontal connections, feedback connections have very large convergence regions and can carry information from long distances in the visual field, achieving the desired goal of integrating global information with local and precise processing.

Areas V1 and V2 could act as *active blackboards* integrating in their neuronal responses of neurons in other higher order areas that are likely to be activated later.

The idea of retroinjection in lower order areas makes it necessary to identify which parameters are processed first, since the results of this first computation can be done sufficiently fast, so that the result of computations done in higher order areas can actually influence the response of neurons in areas V1 and V2. Because they are located close to the entry point of visual information through the LGN it is usually assumed that neurons in the low order areas V1 and V2 do have shorter latencies on visual stimulation than neurons of areas located higher in the processing hierarchy. It was shown recently that neurons in several cortical areas are activated very early. Practically at the same time as areas V1 and V2. It is clear that there are a number of cortical areas, such as MT, MST, the frontal eye field (FEF) and 7a that contain neurons that are activated sufficiently early to influence neurons in areas V1 and V2. These areas are in the parietal cortex or, in the case of FEF, in a region of frontal cortex that is heavily interconnected with the parietal cortex. These areas, often called the *fast brain*, belong to the dorsal stream or its continuation in the frontal cortex.

The main ideas behind this organization are

1. The first activity to reach visual cortex is provided by the M channel Figure 4.2

2. areas of the dorsal stream are activated very fast and are thus in a position to influence the behavior of neurons in areas V1 and V2

3. caused by the low conduction time of feedback connections, results of early computations done through the first processing pass can influence the responses of V1 and V2 neurons just in time for the arrival of the activity of the P stream, which is delayed by 20 ms

4. areas V1 and V2 provide general purpose representations, or *active blackboards* which integrate, within their responses the computations done at hierarchically higher levels.

5. because of the rapid activation of neurons in the dorsal stream, it is likely that they can influence responses of neurons in the ventral stream mainly by retroinjecting information in areas V1 and V2.

These items showed us a strong necessity for recurrent feedback in our attempt to define an artificial neural network model. Bullier derived evidence

for these retroinjecting connections by fMRI recordings he discussed in detail in [Bul01].

4.4 What we Derive from Neuro-Anatomy

We will now briefly recapitulate on the neuro-anatomical findings and principles we want to integrate into our artificial neural network model. The first issue is the hierarchical organization of the system and the increasing of the size of the receptive field the more advanced the layer is. Also the role of retroinjection is taken into account in the development of our system.

What we can derive from this chapter is:

- An artificial neural network has to be constructed out of hierarchically organized processing layers
- Information representation on each level should be similar to the changes in size of the receptive field in the visual areas; the more abstract (higher) the processing layer, the rougher the resolution
- An artificial model has to implement a short-term memory, recurrent connections, and retroinjection connections to realize prediction abilities.
- The dorsal processing stream is associated with motion, the representation of object locations, and the control of the eyes and arms, especially when visual information is used to guide saccades or to reach for an object. It is so to say the stream we have to look at when designing an artificial motion perception model.

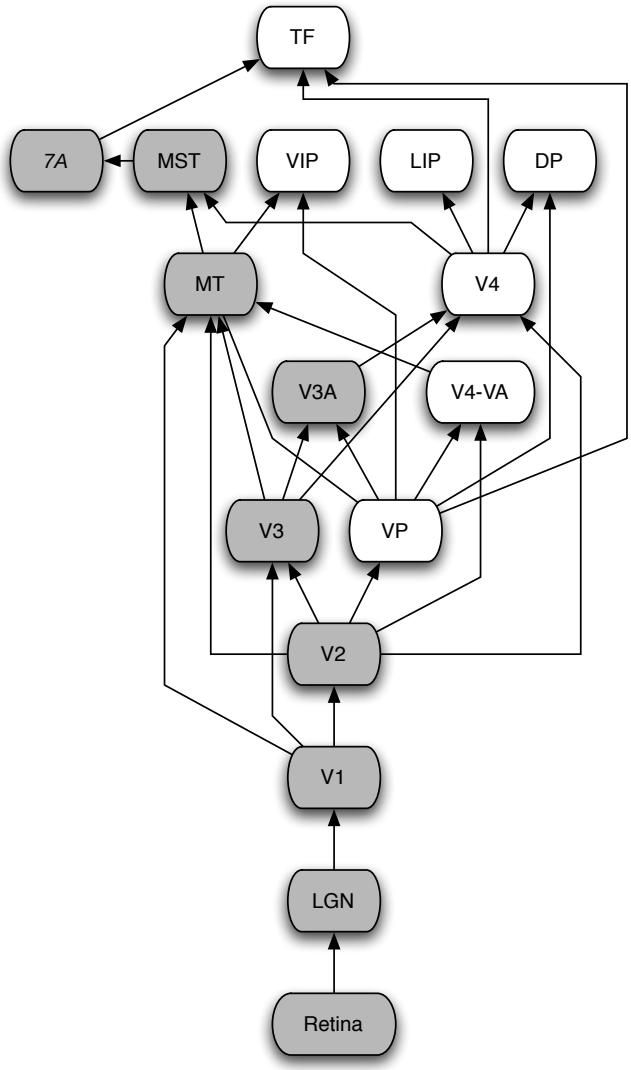

Figure 4.4: Excerpt of the structure diagram of the macaque visual cortex. Shaded areas play a substantial role in motion perception.

Chapter 5

Motion Perception in Cognitive Psychology

Within the research group this thesis has been started, several experiments on spatio-temporal memory and motion processing were conducted. The central question was:

What are the motion primitives and prototypical motion shapes which are used by the visual system in order to classify and predict trajectories?

This question guided the experiments described in this psychophysics related chapter. A number of different experimental paradigms were applied, in which the complexity of the motion stimuli used was increased successively. The experiments started with simple kinks and curves, went further with occluded paths, and ambiguous displays and ended with extended and very complex trajectories. In all experiments temporal parameters were varied in order to gain insights into memory-based processes. The empirical results leading to a motion shape vocabulary of the visual system are transferred to our modeling approaches. These approaches include the consideration of different levels of processing and representation. For the higher, more cognitive level of spatio-temporal information processing a propositional framework for the qualitative representation of motion information has been developed by A. Musto [MSE+00]. It uses linguistic concepts for qualitative motion vectors like left-turn, u-turn, loop, ..., which are correlated to motion primitives found in our experiments. This qualitative approach can be used to describe, generalize and calculate trajectories on different levels of abstraction (described in detail in [Nih73]). However, there remains a gap between the first visual stage of dynamic processing (realized e.g. by an orthogonal access memory stage [SZ95]) and a linguistically coded qualitative representation of motion information on the other side.

The multi-layer neural network model introduced later is the result of our search for a way of spatio-temporal information processing. In order to enable the representation, processing and prediction of spatio-temporal

patterns on different levels of granularity, a hierarchical network model has been investigated which consists mainly of self-organizing map like layers organized in a hierarchical manner.

5.1 Visual Spatio-Temporal Memory

Motion perception models which are based on dynamic concepts and capable of processing spatio-temporal information are quite rare in vision research. Most research was done in the field of static pattern classification. An early motion perception model was dividing the complete system into two visual subsystems each of the subsystems responsible for a single task. One realization of a visual system consisting of two parts, the so called transient/sustained dichotomy concept was described by Watson [Wat87]. One part of the system, the *sustained subsystem*, is characterized by its slow temporal integrating and low-pass filtering processing behavior. The second part of the system, namely the *transient subsystem* simply detects changes in the temporal domain of the provided signal which occurs if objects appear or disappear from an observed scenery.

This rather simple model is not able to completely explain the perception of moving objects, what motivated a refined dynamic perception model: *optic flow*.

Optic Flow is characterized by the displacement of each single point of a scene between two sampling events. So the spatial distance between a single observed point at time t and at time $t + \Delta t$ generates one single optic flow vector. The sum of all these difference vectors define the motion vector field characterizing the optic flow. So we can say that the field assigns to each spatial point of every sampling event in time a velocity vector. Computationally the calculation is a simple value comparison of two points in time and space. Reichardt investigated the flies nervous system to find the mechanisms responsible for processing motion. To process changes in a dynamic scenery e.g. the fly uses a quite simple realization of a motion detector, the so called Reichardt detector [PR73]. Figure 5.1 shows the simplest possible detector which is able to detect motion in a single direction. This simple Reichardt detector uses two spatially displaced inputs provided by retinal photo-receptors to calculate a response. If a stimulus is applied to the first and after a appropriate delay to the second input receptor, the two signals traveling to the detector arrive simultaneously and are summed up to the excitation. The more synchronously the signals arrive the higher the excitation and the higher the output of the detector. It is quite easy to define more sophisticated detectors by combining the inputs by inhibitory lines. A detector capable to react to two opponent stimuli with an inverted output signal is shown in Figure 5.2. Motion detection in a 2D environment demands a model called orientation selective filter Figure 5.3. Higher order

motion approximations were investigated e.g. by Glünder [Glü90], Nomura & Miike, [NM91], and Zetzsche& Barth [ZB91]. Optic flow vector fields are not more than the temporal and the spatial property of a provided input signal and therefore nothing more than the "spatio-temporal orientation".

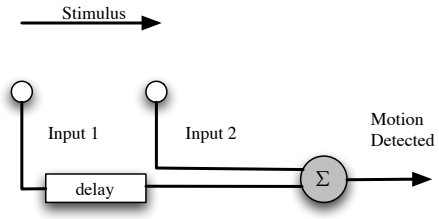

Figure 5.1: Reichardt Detector sensitiv to a single direction

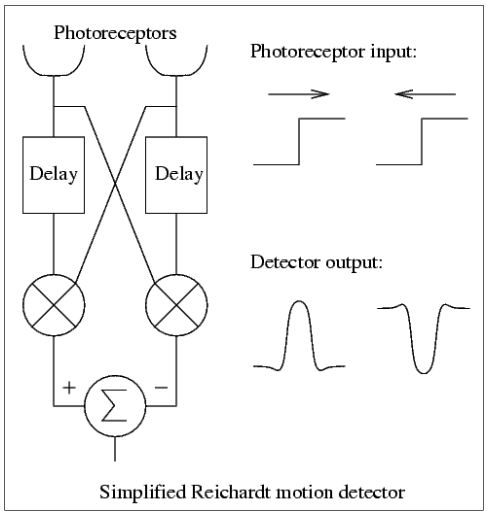

Figure 5.2: Generalized Reichardt Dectector

Common to the before introduced approaches is what Schill and Zetzsche call a *collinear* concept of the representation of time and space [SZ95]. In the models discussed above the internal representation of the spatio-temporal input is a modified projection of the input signal. The temporal properties of the external and internal signal are preserved. Due to the fact, that the systems do some kind of processing like filtering, an additional delay is possible. Regarding one single moment of time –one sampling event– the only internally available property in the system is a local-momentary signal like velocity. The introduced collinear concept of the representation of time is not providing a direct access past system states. To illustrate why the access

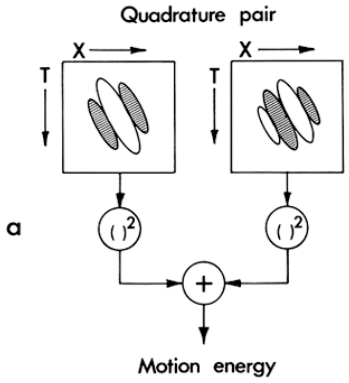

Figure 5.3: Two linear motion filters from a quadrature pair with a peek response at 90° out of phase [AB85]

to past states induce a problem for motion perception we will first have a look on the structural requirements for a static spatial pattern processing system. Obviously the recognition of spatial patterns cannot sufficiently be based on a single local property like the local orientation at a spatial position, since a combination of several local properties is necessary to form a single spatial pattern. So the task of spatial pattern recognition is based on the combination of single local spatial orientations to a static pattern. To realize this processing ability it is necessary to have an instance which has access to spatial features from different locations and to several time steps. The technical realization of the instance which combines those different locations could be realized by the famous grandmother cell or for example by a neural assembly. The temporal properties demand a memory structure. Independent to the distinct realization of the combination instance for the spatial properties it can be stated, that this instance has to have full access to various spatial features from various locations.

In contrast to the temporal domain, spatial properties can be mapped quite easily by interconnecting nerve fibers carrying signals from the retina to the "processing" areas simultaneously.

5.2 Dynamic Spatio-Temporal Memory

Motion queues with larger complexity like the previously mentioned stimuli need a more sophisticated motion processing model. The main question is, how human beings perceive spatio-temporal structures. Structures of the complexity like figures in figure skating, throws in Judo, the characteristics in the way a person walks [Joh75], the recognition of long complicated dynamic gestures as in sign language, or dealing with phenomena arising

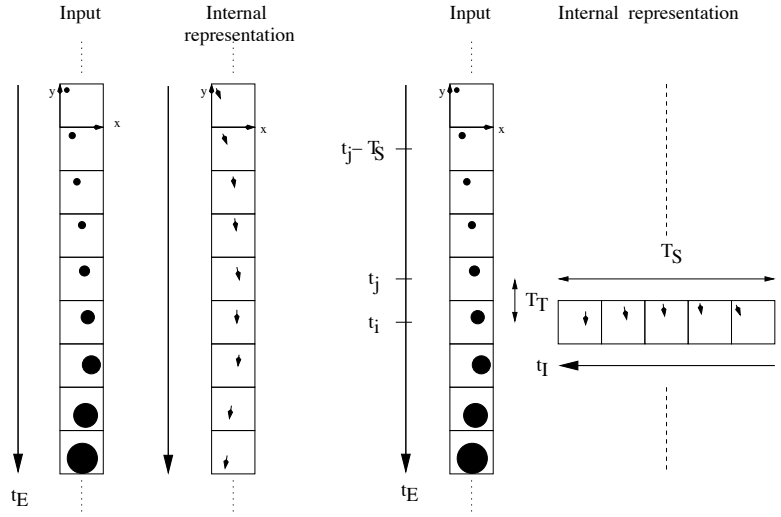

Figure 5.4: Comparison of a collinear representation, shown on the left hand side, and the orthogonal representation of a spatio-temporal input on the right hand side.

through ego-motion in environments densely occupied with obstacles or observing moving objects: occlusion or in other words discontinuities. Imagine a bird flying behind a tree. Trying to track the birds trajectory, our visual system is able to estimate the point of reappearance after the occlusion. This is necessary because of the insufficient speed of occulomotoric system to search for the stimuli after the occlusion again. Also aiming for a moving target makes it necessary to predict the trajectory of the target. Handling occlusions of course can only be realized with a system capable of predicting and approximating motion of objects without any externally provided stimuli.

Detectors like the Reichardt-Detector shown in Figure 5.2 or ensembles of orientation-selective wavelet patches can model the spatial domain of the spatio-temporal memory. In the temporal domain the case is a different one. Because of the necessity of an access to past elements of an observed motion path, to classify the motion correctly and to estimate correct predictions, Schill and Zetzsche [SZ95] introduced an orthogonal-access memory. This memory renders the system able to access various past motion atoms simultaneously with no computational effort. The necessity of such a temporal short-term memory gets clear, if the system has to process motion paths and the input stimulus is characterized by direction and velocity. The system is only able to classify motion, if it is integrating past motion atoms –we define a motion atom consisting of direction and velocity– to a motion segment.

In the spatial domain, access can basically be realized by interconnecting

nerve fibers, providing information exchange channels. Access to the temporal domain, however, requires the transport of information across time which can only be provided by a distinct operation, the *storage* of information. The required array of storage elements can be called a memory. Since we want to deal with the perception of dynamic scenes, with the extraction of information from space and time, the memory has to be a *spatio-temporal memory*.

This memory must not necessarily correspond with a single spatially restricted area of the brain and realized by a specific class of neural elements. It is also possible that the memory is distributed around multiple visual processing areas interconnected with each other massively. The distributed processing of motion information and especially the massive interconnection between those motion processing areas in the brain were recently investigated in neuro-anatomy by J. Bullier [Bul01].

In this thesis we construct a system associating a spatio-temporal memory on an artificial neural network definition but avoiding a pure spike based model realization, in order to deliver a system with applicable computing complexity and still keeping the demands of the model fulfilled. The basic concept of the dynamic memory is the mapping of time to physical properties reflected in an artificial neural network model. Time and the memory state of the system must be accessible simultaneously.

To defend the statement that the concept of simultaneous access not only holds for some special assumption Zetzsche and Schill argued that, unless it might be easier to construct a sequential processing system like a standard computer equipped with a camera subsystem, it is missing simply the orthogonal access. The access problem might be solved by the provision of a memory structure storing some of the last incoming visual information frames. This additional memory supports the hypothesis of an orthogonal access memory as this memory enables the mapping of the incoming information stream into a spatially distributed representation. By integrating such a memory it is only shown that the presence of it is quite convenient but it is so far not shown that it is really necessary. Using the concept of a state machine, you can think of a system which receives input in sequential order and propagates –like a shift register– each new occurring frame into a state. States are subsequently changed as the result of the input. It seems that with this state machine definition the necessity for a mapping of the sequential information into a spatial memory gets obsolete. But state machines are inherently different to systems that provide a direct mapping with direct information access. Nonetheless they are only another solution to the access problem with the the main differences, that they are more indirect. The information about past events is encoded into the states of the system, whereas *state* refers to a different metaphor for a spatially distributed activation pattern.

5.3 Psychophysic Experimental Results

The development of the hierarchical neural network model has also been guided by experimental research and investigations on the representation of spatio-temporal structures on early visual processing levels, related to the concept of ultra short-term memory done in our research group at the Institute of Medical Psychology.

In all experiments described below human test subjects were sitting in a semi-darkened room in front of a computer display (CRT with 100Hz refresh frequency) at a viewing distance of about one meter. As stimulus they saw a black dot moving on a white screen along an invisible path. The setup met real-time capability.

Discrimination of Changing Direction

In the first series of experiments the ability of the visual system to discriminate simple motion trajectories is measured. The stimulus is sketched in Figure 5.5. In each trial a pair of such trajectories was presented subsequently. The subjects' task was to decide whether there was a greater change in the direction of the motion in the first or in the second stimulus pair and to indicate this by pressing one of two buttons. The stimuli were arbitrarily rotated –also within the trials– to prevent the subject from using additional cues, like external reference points.

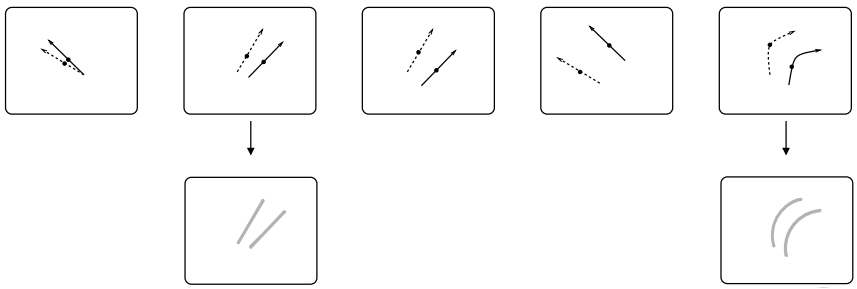

Figure 5.5: Direction discrimination of straight and curved trajectories

Under all conditions the main result was a slight increase in performance up to several hundred msec stimulus duration, followed by a widely invariant performance up to 2.5 sec. This is surprising since the visual impression of a motion that lasts of only a few hundred msec is very different from one lasting 2.5 seconds, a result which is of course taken into account in the neural network model introduced later. The probability of correct decisions was in the range between nearly chance and almost perfect performance. As expected, the more the trajectories differed in curvature the better the performance. A less accurate processing with continuously varying direction

of motion could be observed in comparison to trajectories with an abrupt change in direction. This task-dependent accuracy defines appropriate parameter values and constraints for our neural network model.

Neither relatively fast movements nor long durations have an influence on subjects' sensitivity. This invariance is supported by the experiments on direction and curvature discrimination discussed in [ESR+00]. These experiments with curved motion paths also showed that the discrimination performance was significantly different –better– when corresponding static versions of the dynamic stimuli were used. Therefore the observed invariance can neither be explained by motion-selective mechanisms nor by the assumption of static internal representations in the form of spatial trajectories of the dynamic stimuli.

An analysis of the experimental results on this memory stage has revealed inconsistencies arising with the dominant views of how information is represented and stored on this stage. Introducing a memory structure which provides basic requirements for the processing of spatio-temporal information resolves these inconsistencies. The key feature of the memory structure is the provision of an orthogonal access structure which is achieved by mapping external time and its internal representation, temporal structure into locally distributed activities. This mapping enables a parallel access to a whole sequence of recently past visual inputs by higher level processing stages which is called an orthogonal access memory (OAM) following [SZ95].

Given this spatio-temporal structure one important question is what are the abilities and limitations of the human visual system. To investigate this question various psychophysical experiments to the processing of spatio-temporal information on different time scales were carried out by E. Eisenkolb and F. Röhrbein [ESR+00] [RSB+03]. Human subject performance to spatial and temporal stimulus properties has been measured in a number of different experimental designs and with stimuli ranging from simple straight dot movements up to more complex motion paths with higher-order shapes like loops or kinks.

One series of experiments for example addressed the ability of the visual system to discriminate orientations of straight and curved trajectories both induced by moving dots like sketched in Figure 5.5. The visual system shows a clearly better discrimination performance in case of the induced straight trajectories. The orientation classes obtained are used to determine the classification power of the neural network model. In order to compare dynamic information processing with the static one always the static versions of the moving stimuli was devised. The results suggest that static and dynamic information is processed differently, which is in contrast to the view that simple dynamic patterns are processed by static mechanisms.

In the latest experiments on the generalization performance in dependency on temporal constraints were investigated with the stimuli shown in Figure 5.9. The results will give us hints on the design of the levels of ab-

straction of our neural network model as e.g. given by the number of the processing layers and the granularity with which spatio-temporal information is processed in the model. The experimental design is based on a pilot study in which subjects have to reproduce a complex trajectory shape on a touch screen. The results of the pilot study indicated evidence for the existence of a number of prototypical shapes used by the visual system to classify a trajectory.

Curvature Discrimination

The experiments on direction discrimination didn't exhibit any influence of a temporal separation on the discriminability of simple motion paths. In order to test the inter stimulus interval invariance Eisenkolb et. al. devised another experiment. Instead of direction discrimination the discrimination ability of human subjects on curved trajectories has been investigated. In

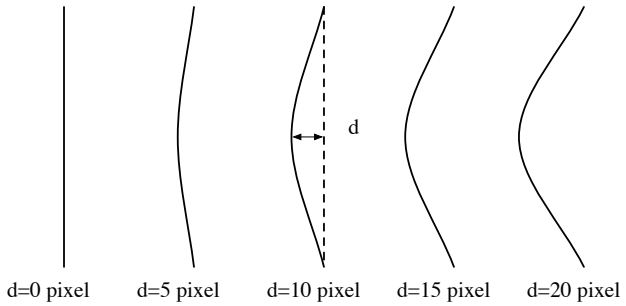

Figure 5.6: Curvature classes for discrimination experiments

the experiments the performance of the test subjects decreased as the degree of curvature was increased. A temporal separation up to 400 msec does not result in an impairment of inner availability of the motion paths. All subjects showed, partly diverging, strategies to cope with the discrimination task.

No memory effect caused by temporal separation could be observed in the experiments presented above. While in the experiments to direction discrimination the explanation for a missing memory effect is that processing occurs too fast for a capacity limitation to become manifest, here a trade off effect can be observed, as subjects tend to dedicate more attention to the second path to the disadvantage of the first path.

Occluded Spatio-Temporal Patterns

To gain further information about the processing of static vs. dynamic patterns an experimental paradigm called the *Dynamic Poggendorff Stimulus* as

shown in Figure 5.7 was developed. A second motivation has been that vision often occurs under circumstances of occlusion. In this sense the dynamic poggendorff stimuli serves as a spatio-temporal completion task. Spatio-temporal binding is one of the most puzzling problems both in psychophysical research and robot vision. The testing subjects saw a single black dot moving

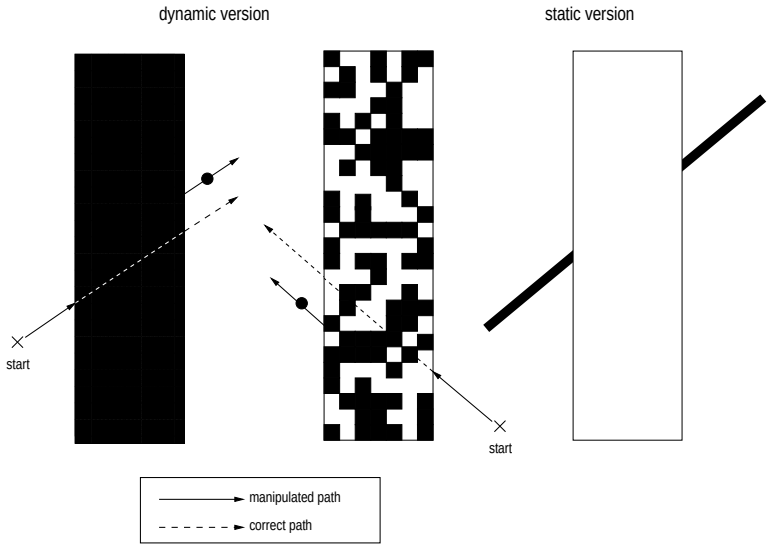

Figure 5.7: The Poggendorff Illusion

along an invisible straight oblique line. The middle segment of the motion path was occluded by a vertical bar dividing it into two visible segments. This setup is a dynamic version of the *Poggendorff* illusion. The subjects were instructed to adjust the position where the moving dot reappeared in order to bring both visible segments into alignment. The parameters which where varied are: the dot velocity, the width of the occluding bar and the filling texture of the occluding bar. All permutations of the conditions where tested.

The filling mode has no influence on the alignment error at all. An occluding bar leads to a positive alignment error, in analogy to the static Poggendorff experiment. Increasing the bar width leads to an increase of the alignment error by a factor of 2.3. Increasing the dot velocity leads to a decrease of the alignment error by 0.9.

It is an interesting result that the presence of an occluding bar leads to a systematic positive alignment error. The dynamic Poggendorff illusion thus clearly suggests a qualitative analogy between static and dynamic processing modes. Research reported in [EZM$^+$98] on the other hand suggests a drastically quantitative difference of the processing of static and dynamic stimuli. It is also likely that the task employed involves memory based mechanisms

important for prediction of courses of motion. What on the other hand does by no means explain why there is a dynamical Poggendorff illusion.

Spatio-Temporal Prototypes

The experiment on spatio-temporal prototypes addressed the issue of motion prototypes presenting complex motion trajectories. The investigations on complex motion started with a piloting study (described in detail in [RSB+03]) in which subjects had to reproduce a complex motion shape on a touch-screen (Figure 5.8) by sketching the trajectory with one finger on the surface of the touch-screen. No visual feedback was provided. Examining

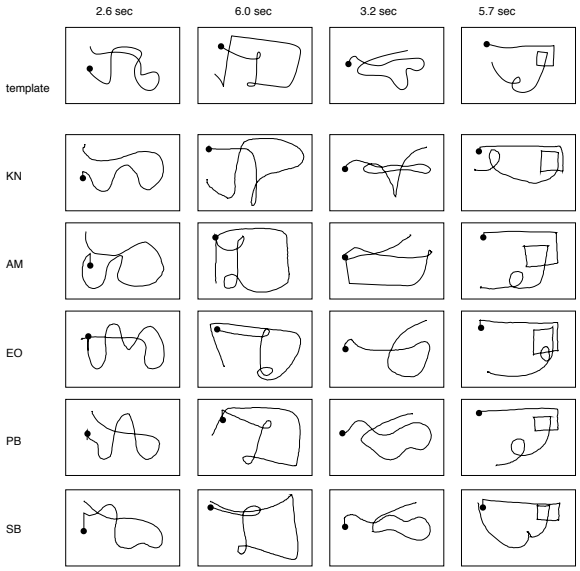

Figure 5.8: Trajectory reproduction task. The top line shows the templates. The stimulus is the black dot moving along the sketched trajectory. Each other line a single subjects reproduction, by drawing the trajectory with a finger. The path was not visible neither in the template case nor in the reproduction phase.

the stimuli and the reproduced trajectories in [RSB+03] it can be derived that accuracy is mainly determined by implicit features of the motion paths. These implicit features point towards the existence of prototypical motion shapes. Encouraged by the results from the piloting study, an experiment with stimuli simple enough to allow for quantitative results and complex enough, so that conclusions about the existence of prototypes can still be drawn, was conducted. Stimuli which can be described as a sequences of qualitative motion vectors (QMVs) [MSE+00] seem to be especially suited

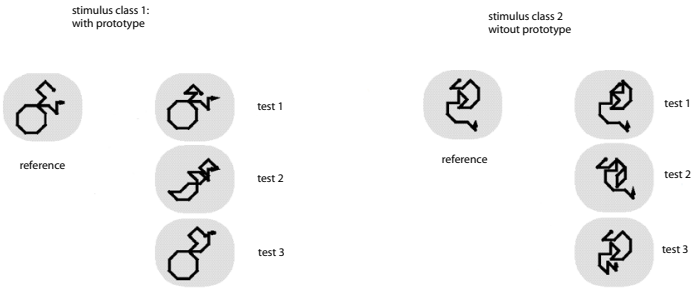

Figure 5.9: Trajectories of QMV stimuli

for such an experiment. Several trajectories used in the experiments are shown in Figure 5.9. They consist of 16 segments of constant length. Two of these restricted trajectories where presented as a first pair and after a short pause another two as a second pair (Figure 5.9). The subjects' task was to indicate whether there was a difference in the dot movement presented in the first pair, or in second pair. The experiment consists of two types of trajectories. The first class has a prototypical motion shape the second none (Figure 5.9). Whereas the prototypical shape in the first class is the quasi circular part of the trajectory.

The main result of the QMV experiment is that the discrimination performance strongly depends on the existence of motion prototypes. Looking at the results for the second stimulus type, which is the one without a motion prototype, supports the idea of prototype processing. Testing the stimulus of the second class, which has no prototypical element and the same amount of motion changes, results in a very poor discrimination performance. For this class of stimuli only the third variation labeled test 3 in Figure 5.9 leads to a performance comparable to the first stimulus class. This can be explained as the modified stimulus contains a zigzag-movement which might be processed also as a prototypical element. In order to reveal more elements of this *shape vocabulary* experiments with a number of supposed prototypes could be investigated. This experimental framework might also be extended to address explicitly the processing on different time scales and on several spatial granularities, e.g. by systematically varying the moving time, number of segments, and possible orientations. But as such experiments where not done so far, we are forced to take a different school into account to gather hints for the construction of our model.

5.4 What we Derive for our Model

Since the spatio-temporal memory is one essential concept of this thesis, it deserves a recapitulation of our arguments on a slightly different level.

If information should be detected or identified by some computational structure, this structure must have access to necessary information. Just as the recognition of a static spatial pattern needs to be based on access to several spatial locations of the image, the processing of spatio-temporal patterns requires access to several spatio-temporal *locations* (locations in space *associated* with time steps). This access can only be obtained, if at a given time step in external physical time a physical representation of the presented events exists which occurred during some recent period. The required representation can be called a *dynamic spatio-temporal memory*. The basic function of this memory is the *mapping of time into space*. In contrast to the *collinear* concept, this constitutes an *orthogonal* concept of the representation of time. It is only possible by a memory with orthogonal access, rendering a system capable to extract information about the spatio-temporal structure of its environment. The experiments not only came up with important constraints for our neural network model. They also provided us with hints and proof of concept about the structure of the model.

- Processing in different granularities was proven by the experiments on spatio-temporal prototypes.
- How information is transported from one layer to the next higher processing layer is realized by a kind of temporal sub-sampling.
- The Dynamic Poggendorff illusion constrained the prediction capabilities.
- All experiments provided temporal constraints.
- The necessity of an orthogonal access memory in our model was supported by the experiments.

Chapter 6

Neural Networks for Static Patterns

Before we can take the next step to discuss sequence processing models, we have to introduce some static pattern processing models. These models are the foundations of the multi-layer model for spatio-temporal sequence processing introduced later.

6.1 Multilayer Perceptron

The first class of neural networks introduced in this work is the multilayer perceptron (MLP) with its well known *error back-propagation* learning algorithm introduced by Rumelhart, Hinton and Williams (1986). It can be said that this algorithm has redeemed the research in this area from a long stasis inferred by a book of Minsky and Papert [MP69]. As the name of the model describes, MLPs consists of several layers of perceptrons. The first layer is a set of sensory units also called input layer, followed by one or more hidden layers of computational nodes. The output is also formed by a layer of computational nodes. One important limitation is the character of the transfer function of the hidden and output layer. This function has to be *nonlinear*, otherwise the operations between the layers would degenerate to a simple dot product, and the multi-layer network would only have the computing capability of a single layer network, restricting the class of problems solvable with the network to linear separable problems. Often a sigmoidal function is used as activation function. A MLP consists of three major elements:

- A nonlinear activation function e.g.:

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \tag{6.1}$$

  whereas $\sigma(x)$ denotes the activation function applied to the input x

- One or more *hidden* layers excluding the *input* and *output* layer
- Fully inter-connection of the neurons

Error back-propagation is working with two processing phases. In the first phase a pair of training sample and associated output vectors are presented to the network. Then the error (e.g. the euclidian distance) between actual output and training signal is calculated. In the second step, the mapping error is propagated back through the network while adapting the weight vectors of the passed neurons by a gradient descent algorithm. The rate of adaptation is a decreasing function of time. Several precautions have to be looked at, for example local minima and learning rate.

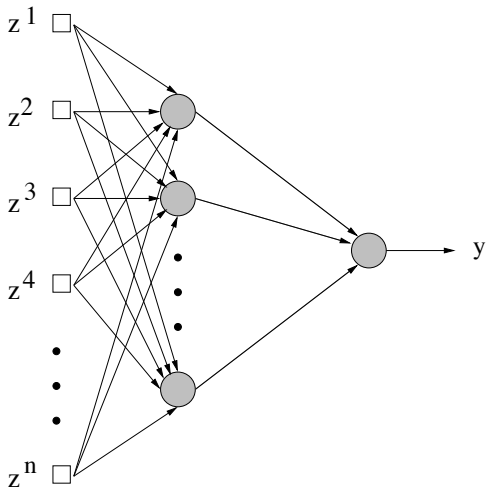

Figure 6.1: Structure of a multi-layer perceptron network

As you can find a description of back-propagation in every good lecture book and as we do not introduce any novelties to back-propagation based MLPs we will refer the interested reader to textbooks like [Hay94] or [Roj96].

6.2 Self-Organizing Maps

Self-organization as a biological optimization process was realized through several algorithms. Most attention however was addressed to Kohonen's proposition of an discrete approximation of the biological process of self-organization. Kohonen's self-organizing maps (SOMs) as a computational technique turned out to be useful for a wide range of applications. The SOM algorithm is quite efficient, due to the lack of the necessity of iterative excitation level determination which is replaced by a single operation maximum detection or a single operation euclidean distance calculation, on the

other hand this simplification brings us away from biology; we will get back to this point later on as we want to make our motion perception model more realistic.

A neuro-physiological background of self-organizing maps are the sensorial maps of the human cortex. If we look at the sensory cortex, spatial closely situated receptors project to closely spatial organized cortical areas. The mapping of the somato-sensory system is often illustrated with a homunculus like in Figure 6.2. So one can state that the topological representation is preserved. Despite of the fact, that the cortex containing 10^4 synapses per neuron is massively interconnected, the brain reacts to external stimuli always with a local activity which can be seen as an argument for the winner take all scheme used within the SOM algorithm. With modern imaging technology it is possible to assign these local activities to functional areas. The cortex can be subdivided in several functional areas. The visual cortex is one of the functional areas. These properties show, in our opinion, that SOM like models provide a good abstraction of single layers of real brains.

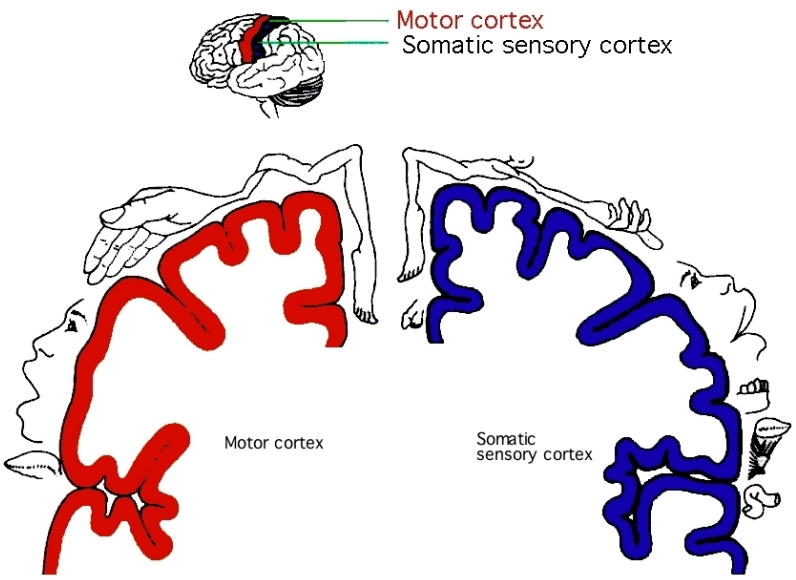

Figure 6.2: Somato-sensory and Motor Homunculus (W. Penfield)

6.2.1 Maximum Maps

The original definition of Kohonen's algorithm has many similarities to the biological model it was based on. The organizing structure of a SOM can be defined by a two-dimensional lattice with neurons placed on its junction

$x(n)$	Binary input/training vector n
X	Training set
$y_i(n)$	output neuron i
$y_b(n)$	output best matching neuron
r	two dimensional position vector
w_i	Weight vector neuron i
$N_{ib}(n)$	Neighborhood function
$\gamma(n)$	Learning rate
$\|\| \cdot \|\|$	Euclidian distance

Table 6.1: Symbols Self-Organizing Maps

points. Other structures like a torus are also possible but will only result in better network performance, if the generative process of the statistic of the available training vectors will form a torus like distribution. Therefore and due to the fact that the brain is not torus shaped we will discard these structures here. In some cases a neighborhood of six neurons –hexagonal neighborhood lattice vs. rectangular– will result in better network performance, again it depends on the statistics of the provided data. The neurons are defined by their two-dimensional position vector $r_i \in A$ which determines the position on the lattice, and a weight vector w_i with the same dimension like the input/training vector.

The output value y_i of each neuron, the excitation function is determined by the product of input value x_j with each neurons weight vector w_i similar to traditional feed-forward networks:

$$y_i(n) = \sum_j x_j(n) w_i \tag{6.2}$$

This is done iteratively for every input vector. The best responding neuron is determined by the search for the maximum activated neuron:

$$y_b = \max_i(y_i) \tag{6.3}$$

The weight adaptation in the maximum map is constrained by the learning gain $\gamma(n)$ which is a decreasing function of time supplying values between $[0:1]$

$$w_i(n+1) = \frac{w_i(n) + \gamma(n) N_{i,b}(n) x(n)}{||w_i(n) + \gamma(n) N_{ib}(n) x(n)||} \tag{6.4}$$

and a neighborhood kernel N_{ib} centered by the winning neuron. The simplest form of a neighborhood kernel is the bubble neighborhood. Neurons within the neighboring range are adapted equally strong, neurons to far away are not adapted at all.

$$N_{ib} = \begin{cases} 1 & : \quad i \in N_b(n) \\ 0 & : \quad \text{otherwise} \end{cases} \tag{6.5}$$

A gaussian neighborhood kernel is very common, too:

$$N_{ib}(n) \quad = \quad \exp\left(-\frac{||r_i - r_b(n)||_2^2}{2\sigma(n)^2}\right) \tag{6.6}$$

The training algorithm is unsupervised. Randomly chosen p-dimensional input vectors $x(n)$ are drawn out of the set X and repeatedly presented to the net. The weights are adapted with the above defined weight adaption rule after the presentation of each input pattern. This is done for all input patterns several times, while the learning rate $\gamma(n)$ and the neighborhood $N_{ib}(n)$ decrease with every training iteration.

Finally, the major disadvantage of maximum maps is the slow convergence and the huge amount of training cycles to achieve a good mapping of the input values to the self-organizing map.

6.2.2 Minimum Maps

This variant was proposed first by Ritter and Kohonen [RK86]. It introduced a different neuron concept referring the maximum map model. The minimum maps is searching for the minimum difference between input and weight representation. The training algorithm of the minimal map is of course again based on an unsupervised scheme. Randomly chosen p-dimensional input vectors $x(n)$ are drawn out of the input data set X and presented to the net. The adaptation of the neurons' weight vectors $w_i(n)$ at training step n to the input depends on their euclidian distance $||\cdot||_p$ to the current input vector $x(n)$. The output of a neuron is $y_i(n)$:

$$y_i(n) \quad = \quad ||x(n) - w_i(n)||_p \tag{6.7}$$

One of the neurons having the smallest distance to $x(n)$ at iteration step n is called best matching neuron, its distance is denoted by $y_b(n)$.

$$\begin{aligned} y_b(n) = & \quad ||x(n) - w_b(n)||_p \\ = & \quad \min_{i \in M} \{||x(n) - w_i(n)||_p\} \end{aligned} \tag{6.8}$$

To generate a topology preserving representation of the input vectors, similar input vectors have to be represented by neurons with a similar weight vector. Therefor a neighborhood function is introduced. The adaptation of the weights $w_i(n)$ of the neurons addressed by the position vector r_i in the neighborhood of the best matching neuron at position $r_b(n)$ decreases according to the neighborhood function:

$$N_{ib}(n) \quad = \quad \exp\left(-\frac{||r_i - r_b(n)||_2^2}{2\sigma(n)^2}\right) \tag{6.9}$$

$\sigma(n)$ is used to reduce the region around r_b in which the adaptation is strong. $||\cdot||_2$ is the distance defined on the two-dimensional SOM lattice. It is also possible to use other neighborhood kernels like a cut gauss kernel or a simple bubble function. Now having detected the relevant neuron patch, we have to move the selected weights towards the given input stimulus. Figure 6.3 shows a final weight distribution after training the map. One can easily see that similar values are located in similar areas. The weights of the neurons

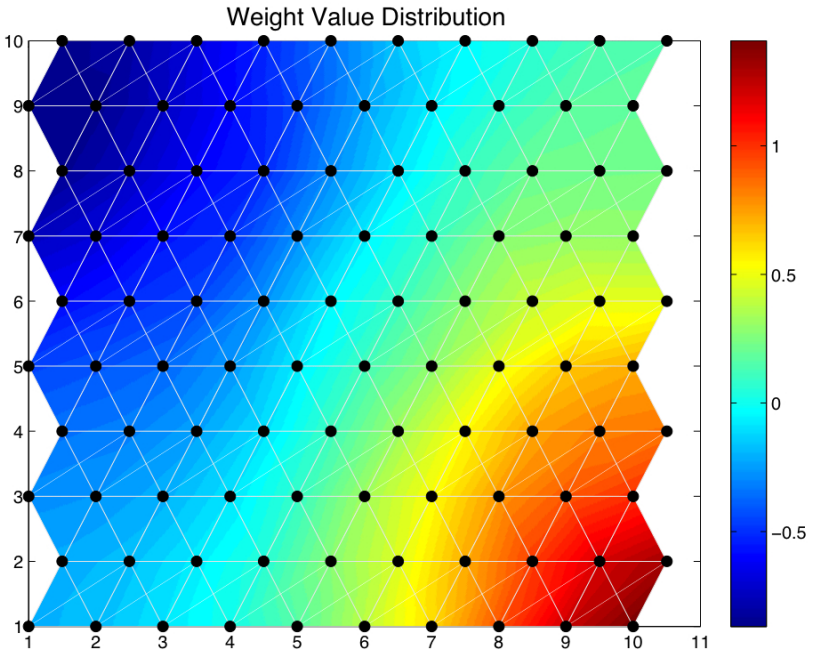

Figure 6.3: Distribution of weight values in a SOM layer.

are adapted for learning step $(n+1)$ through:

$$w_i(n+1) = w_i(n)\gamma(n)N_{ib}(n)((x(n) - w_i(n)) \tag{6.10}$$

whereas the value of the adaptation is dependent on the neighborhood kernel and learning rate. By varying $\gamma(n)$ from $0 \leq \gamma(n) \leq 1$ we can regulate how far the weight vectors are moved in one adaptation step towards to the current input vector. It should be emphasized, that this adaptation phase is not dependent on an error measure like input value output value distance – the system is unsupervised. Usually adaption is done for a predefined number of iterations.

Algorithm 1 General SOM training algorithm

```
Set starting value γ(n), σ²(n)
Create neuron population mesh grid
Random initialization of neuron weights
Randomize Input Data
for (n= 0:max_trainiteration) do
  for (j = 0:number_inputvectors) do
    for i = 0:M do
      y_i = ||x(n) − w_i(n)||
    end for
    y_b = min_i ||x(n) − w_i(n)||
    w_i(n + 1) = w_i(n)γ(n)N_ib(n)((x(n) − w_i(n))
  end for
end for
```

6.3 Neural Associative Memory

Associative Memories are effective and robust storage structures. We can subdivide the known models in two main categories. One class, the Linear Associative Memory (LAM), is processing scalar valued vectors and is trained by a gradient descent based algorithm. The binary value processing models are trained by a so called *clipped* hebbian learning rule. This *one-shot* learning scheme is very efficient. Further differences can be found in the behavior on the provided test data. The LAM reacts with an approximative answer to a noisy signal, whereas the BAM acts like a pattern completion mechanism, filtering the noise of the signal.

x	Binary input vector
y	Binary output vector
t	Binary training vector
w	Weight
μ	Enumerator pattern pairs
M	Number of pattern pairs
n	input vector dim./ number of input neurons
k	output/training vector dimension

Table 6.2: Symbol definitions: Neural Associative Memory

6.3.1 Linear Associative Memory

Given a set of pattern pairs (x^μ, y^μ) with $\mu = 1, 2, \ldots, M$ .... we want to establish a good mapping of $x^\mu \rightarrow t^\mu$ within a linear hetero associative

memory. A linear hetero associative memory is defined by a layer of n linear neurons which perceive an k dimensional real valued vector $x \in \mathbb{R}^k$ as input and an output determined by $y_j = z_j \sum_{i=1}^{k} x_i c_{ij}$. For the sake of completeness: an auto associator learns the mapping of two identical vectors $(x(t) \rightarrow x(t))$ and is used for pattern completion. Obviously it is the aim to adapt the weights c_{ij} in a way that minimizes the difference between output $y \in \mathbb{R}^n$ and training signal or input $t \in \mathbb{R}^n$ of the network. This difference can be described by the quadratic distance measure

$$Z(C) = \sum_{\mu=1}^{M} \sum_{j=1}^{n} (x_j^\mu - y_j^\mu)^2 = \min . \tag{6.11}$$

which can be minimized by the following delta rule:

$$\Delta c_{ij} = \sum_{i=1}^{\mu} (y_j^\mu - x_j^\mu) x_i^\mu \tag{6.12}$$

It is also possible to determine the optimal connection matrix C for the objective function analytically through the partial deviations:

$$\frac{\partial Z}{\partial c_{qr}} = 2 \sum_{\mu=1}^{M} (x_r^\mu - y_r^\mu) = 0 \tag{6.13}$$

In the later introduced spatio-temporal processing model, linear neural associative memory networks are not suitable. The fact that the linear associative memory has a *approximative* response to a given distorted input makes this model unsuitable for our purpose. Therefore we skip to the hetero associative memories for binary patterns. These memory structures have a more *classifier*, pattern completion like behavior.

6.3.2 Hetero Associative Memory for Binary Patterns

Through the approximative response of a linear neural associative memory, the output of the associative memory is inaccurate. If we want to get an exact answer to a e.g. signal with gaussian noise, or if we want signal completion behavior, the appropriate model is a binary hetero neural associative memory introduced below in this section.

The binary pattern processing neural associative memory is based on the *Steinbuch learning matrix* Figure 6.4 which was not defined as a neural net originally. Steinbuchs learning matrix takes two patterns x and y which are encoded as binary vectors. Storing a pattern can be realized in two ways. One way to change the entries is to increment the value c_{ij} of the matrix each time $x_i = y_j = 1$ holds. The second way is to set the matrix once to 1. This scheme is referred as *clipped* learning rule. A similar learning rule is also used for the associative memory introduced below.

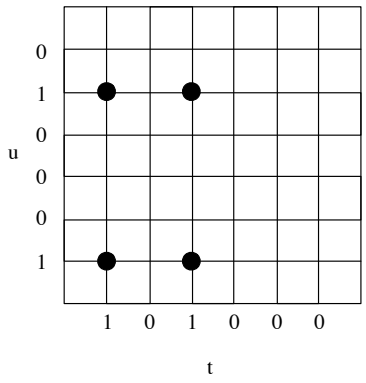

Figure 6.4: Steinbuchs' Learning Matrix

For processing binary patterns we assume that: $x^\mu \in \{0,1\}^k$ and $y^\mu \in \{0,1\}^n$. Like in the preceding definition of a linear associative memory, the input vector x^μ is associated to the proper output vector y^μ to learn an association. In case of performing a data retrieval with a given x^μ as input, the network should provide y^μ as the output. This method of storing an association of two different vectors is called hetero association. The conditions for the patterns are: $x^\mu \neq y^\mu$ for a single μ. The memory consist of a single layer network with n threshold neurons. Each neurons j is computing its dendritic potential to the input $x \in \{0,1\}^k$

$$x_j = \sum_{i=1}^{k} c_{ij} x_i^{(y)} \tag{6.14}$$

and compares these with its threshold value θ_j. The neurons output is then determined by:

$$y_i = \begin{cases} 1 & : \quad x_j \geq \theta_j \\ 0 & : \quad x_j < \theta_j \end{cases} \tag{6.15}$$

Most scenarios use fixed threshold values $\theta_j = \theta$ for all neurons j. In our prediction memory we use an adaptive threshold handling. The threshold is decreased until an output can be retrieved.

In contrast to other neural network models, associative memories are trained with a one shot learning scheme. The learning rule is more like the storage of the input/output vector association in a matrix. Adaptation of the synaptic weights is done by the hebbian learning rule:

$$\Delta c_{ij} = x_i^{(y)} \mu x_j^{(y+1)} \tag{6.16}$$

The synaptic connections are calculated by:

$$c_{ij} = \sum_{\mu=1}^{M} x_i^\mu y_j^\mu \tag{6.17}$$

The learning rate is due to the fact that this is a one shot learning scheme, set to 1. In contrast to the *additive* learning rule above, we use the so called *clipped* learning rule. The difference is simply the limitation of the synaptic connections to a predefined interval with $F : \mathbb{R} \rightarrow \mathbb{R}$.

$$c_{ij} = F\left(\sum_{\mu=1}^{M} x_i^\mu y_j^\mu\right) \tag{6.18}$$

In this case F is defined as a constrained, monotone growing function. For example:

$$F(x) = \begin{cases} x & : \quad x \in [a, b] \\ a & : \quad x < a \\ b & : \quad x > b \end{cases} \tag{6.19}$$

By constraining the additive hebbian learning rule to the range $\{0, 1\}$ one gets the clipped learning rule:

$$\begin{aligned} c_{ij} = & \quad \min\left(\sum_{\mu=1}^{M} x_i^\mu y_j^\mu\right) \\ = & \quad \max_\mu x_i^\mu y_j^\mu \end{aligned} \tag{6.20}$$

Whereas the calculation of the dendritic potential to the provided input x and the threshold rule for the determination of the output values y_i simply keep the same as described above in Eq. (6.14) and Eq. (6.15) respectively.

As we need a pattern completion behavior and no approximative behavior of the associative memory we use a hetero binary neural associative memory. The most puzzling point in using this associative memory is in our opinion the right choice of the input and the output encoding, or in our case better named with transcoding. As we want to use the associative memory to represent an indexing structure for retroinjecting connections based on the activation level of several neurons, we have to develop a transformation of real valued activation values to a proper binary representation. We will come back to the transcoding in place.

Chapter 7

Short-Term Memory in Artificial Neural Networks

Mike Mozer reviewed a collection of possible realizations of a short-term memory for neural networks in [WG94]. The specification of memory models is not complete, whereas the principle how a possible short-term memory realization is often based on one of the introduced filter mechanisms borrowed from adaptive filter theory and signal processing theory. A wider range of adaptive filters and their detailed discussion can be found in [Hay02]. First we should define the term short-term memory:

Definition: *A short-term memory is storing the occurrence of an event in a real valued symbol. This symbol might be manipulated over the time to shape the temporal distance to the occurrence of the stored event.*

We want to realize short-term memory by a recurrent mathematical activation description which is time-dependent. A short-term memory is characterized by its output behavior to an excitation with a standardized input value of 1 within the next following time steps. The next sections will introduce some functions we want to refer as memory kernels, followed by a short discussion on the properties of memories.

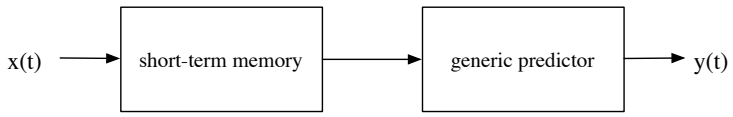

Figure 7.1: The most abstract definition of a system capable of temporal prediction. $x(t)$ is the actual input of the system at time step t whereas $y(t)$ denotes the prediction of the system given a sequence of past input values: $x(1) \dots x(t)$

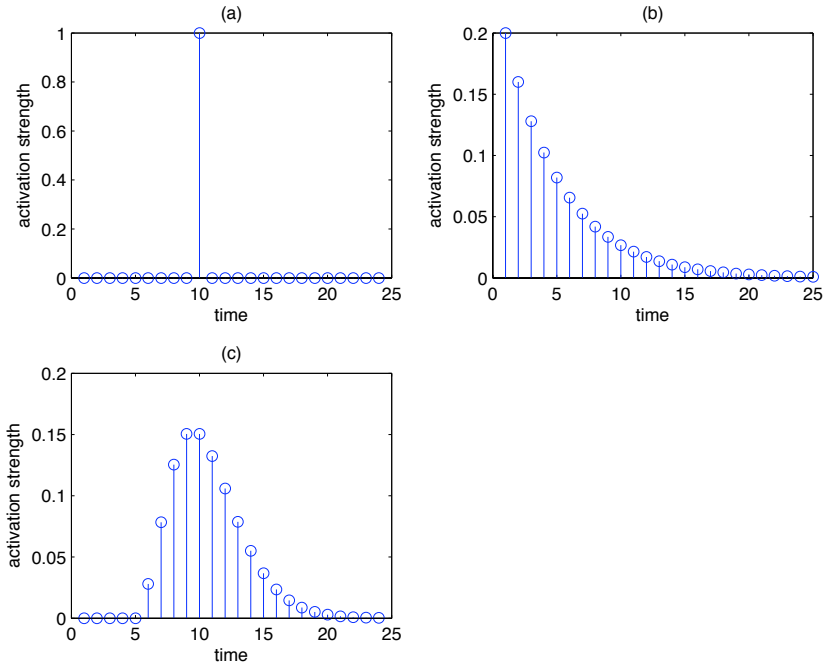

Figure 7.2: Memory kernel functions: (a) tapped delay-line memory with $\omega = 10$. (b) exponential trace memory with $\alpha = 0.8$. (c) gamma trace function with $\omega = 6$ and $\alpha = 0.4$.

7.1 Tapped Delay-Line Memory

A simple way to form a memory is a shift register or in other words a queue-like buffer, which stores in an ordered fashion the n last inputs. This kind of memory is if often labeled *tapped delay-line* model, as the buffer can be formed by a series of delay lines Figure 7.3. In some papers it is also referred as *delay space embedding* and defines the basis of statistical autoregressive models. Tapped delay-line memories are quite common in temporal processing artificial neural networks. Dynamic self-organizing maps are for example a neural network model utilizing tapped delay line memory. We will introduce dynamic self-organizing maps as an example for tapped delay-line memories later.

Tapped delay-line memories Figure 7.2(a) have a predefined capacity of time steps which can be stored in its structure. Given a sequence $x_1(1) \dots x_n(t)$ consisting of a total of ω elements, these memories generate a state representation, where $x_i(t) = x(t - i + 1)$.

We can extend this model simply by introducing a nonuniform sampling rate of past values by specifying delays which are adaptable such that $x_i(t) =$

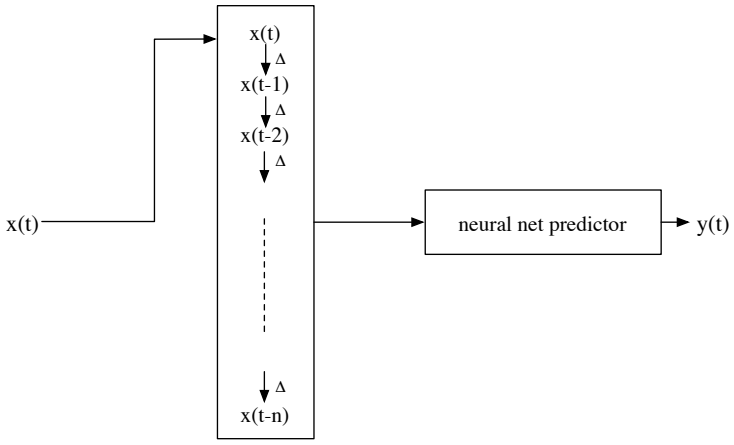

Figure 7.3: Tapped Delay-Line Memory

$x(t-\omega_i)$, ω_i defines the integer-valued delay which is associated with each component i.

To provide a basis for a comparison between the memory models introduced in this chapter, it is necessary to present another formalism for characterizing the delay-line memory which is broad enough to allow an easier comparison with the other forms of memories we discuss later on. Assuming each $x_i(t)$ is a convolution of the input sequence with the kernel function c_i:

$$x_i(t) = \sum_{\tau=1}^{t} c_i(t-\tau)x(\tau), \tag{7.1}$$

for delay-line memories we use:

$$c_i(t) = \begin{cases} 1 & , \quad if \qquad t = \omega_i; \\ 0 & , \quad \text{otherwise.} \end{cases} \tag{7.2}$$

Figure 7.2 illustrates the output of some kernel functions. Through the substitution of different kernel functions, we obtain the forms of memories we will discuss in the next sections.

7.2 Exponential Memory

The functional core of an exponential trace memory is defined by the kernel function

$$c_i(t) = (1-\alpha_i)\alpha_i^t, \tag{7.3}$$

whereas α_i is defined within the interval $[-1, 1]$ Figure 7.2(b). This form of memory has been studied by a wide variety of researches with the focus

on artificial neural networks, but its origin is adaptive filter theory. The exponential trace memory is the simples possible definition of an infinite impulse response filter (IIR). In contrast to the tapped delay-line memory, the exponential trace memory has a soft exponentially vanishing of activation. Tapped delay-line memories simply drop to off after they where activated in the preceding time step. Assuming every neuron in a population is defined to have an exponential decay memory. The neurons in this network can then generate a temporally ordered activation pattern. More recent inputs will always have greater activation strength than temporally more distant inputs.

The response of exponential trace memories x_i can be computed incrementally by:

$$x_i(t) = (1 - \alpha_i)x_i(t) + \mu_i x_i(t-1) \tag{7.4}$$

with the boundary condition $x_i(0) = 0$. Analogue to the delay-line memory, the exponential trace memory consists of Ω state vectors ($x_1(t)$, $x_2(t)$, $x_3(t)$, $\ldots x_\Omega(t)$). The α_i controls the steepness of the exponential trace and therefore indirectly the capacity of the memory.

7.3 Gamma Memory

Before we go on describing the gamma memory, we have to define a characterization of memories *Depth* and *resolution* following [dVP92] and [PdVdO93]. Whereas *depth* refers to how far into the past the memory stores information relative to the memory size. *Resolution* refers to the degree to which information referring a singular element of the input sequence is preserved.

Low-depth memories only hold recent information whereas high-depth memories are able to hold information distant in the past.

High-resolution memories are able to reconstruct the actual element of the input sequence, whereas a low-resolution memory stores more coarse information about the sequence.

Referring to these characterizations of memories, the delay-line memory realizes a low depth with high resolution, and the exponential-trace memory is high depth but low resolution.

The continuous time definition of the gamma memory kernel is defined by a gamma density function as the name *gamma memory* already suggests. Although the original definition of the gamma memory from Principe et.al. primarily deals with continuous time dynamics, it is obviously possible to provide a discrete equivalent of the gamma density function, namely the negative binomial. We can use the negative binomial as a discrete-time replacement to form a discrete-time gamma memory:

$$c_i(t) = \begin{cases} \binom{t}{\omega_i}(1-\alpha_i)^{\omega_i+1}\alpha_i^{t-\omega_i} & \text{if} \quad t \geq \omega_i; \\ 0 & \text{if} \quad t < \omega_i. \end{cases} \tag{7.5}$$

ω_i is again an integer delay factor and α_i is defined in the interval $[0, 1]$. Fig. 7.2(c) shows the output of an exemplary gamma kernel function. Setting $\gamma_i = 0$, the gamma kernel collapses to an exponential-trace kernel. Pushing the limit of α_i to 0, simplifies the gamma-kernel to a delay-line kernel. In signal theory an alternate name for gamma filters is used: adaline. A quite excessive description can be found in [Hay02].

Like the exponential-trace kernel, the convolution of the gamma kernel with the provided input sequence can be computed in a incrementally fashion. However, to compute x_i given α and ω, denominated as $x_{\alpha,\omega}$, it is also necessary to compute $x_{\alpha,j}$ for $j = 0 \ldots \omega - 1$. The recursive form of the equation is

$$x_{\alpha,j}(t) = (1 - \alpha)x_{\alpha,j-1}(t - 1) + \alpha x_{\alpha,j}(t - 1) \tag{7.6}$$

with the boundary conditions

$$x_{\alpha,-1}(t) = x(t + 1) \quad \forall t \geq 0, \text{and} \tag{7.7}$$

$$x_{\alpha,j}(0) = 0 \quad \forall j \geq 0. \tag{7.8}$$

The largest possible ω denoted by Ω must be provided by the memory designer. $\Omega+1 : x_{\alpha,0}, \ldots x_{\alpha,\Omega}$ defines the total number of possible state vectors. Having Ω fixed, the tradeoff between resolution and depth is achieved by triggering α. Large α results in a high-depth, low-resolution memory; a small α yields low-depth, high-resolution memory.

Each set of $\Omega + 1$ state vectors bound to a given α forms a *gamma family*. A gamma memory can consist of many such families, with family i characterized by α_i and Ω_i. This leads to a large and quite complex state representation. If we have some assumptions about the application domain it is possible to reduce the state representation, e.g., the subset $\{x_{\alpha_i,\Omega_i}\}$. This can be seen as a direct extension to the exponential trace memory, which is of the form $\Omega_i = 0 \forall i$. The gamma memory provides us a more realistic onset behavior of the activation of a neuron. Its shape is much more similar to a realistic activation potential due to the fact, that the activation value does not jump directly to its maximum. In biology the activation potential is generated by a ion exchange process through ion specific channels. The channels need of course some time to open also the ion exchange has a small inherent delay. These properties make the gamma memory a bit more biological plausible.

These three memory models discussed until now, are of course not the only thinkable models.

7.4 Various Memory Realizations

Any kernel function generates a distinct memory form. A Gaussian kernel for examples can also be used to realize a symmetric memory around a defined

point in time which defines the point of maximum activation. The gamma memory and its special cases are useful because it is possible to compute their resulting activation by an incremental update procedure. Gaussian memories on the other hand require recursive evaluation of the convolution of the kernel with the input sequence at every subsequent time step. This class of convolutions has same disadvantages in computation time and space.

Radford Neal has suggested a class of polynomial kernels that are defined over a fixed interval of time starting at a fixed point in the past; i.e.,

$$c_i(t) = c_i(t) = \begin{cases} \sum_{j=0}^{\Omega} \alpha_{ij} t^j & , \quad \text{if} \quad t_i^- \leq t \leq t_i^+; \\ 0 & , \quad \text{otherwise.} \end{cases} \tag{7.9}$$

He developed an incremental update rule for gamma memories based on the above kernel. The number of memory state vectors, Ω, determines the order of the polynomial. Update time is independent of the interval width $t_i^+ - t_i^-$. This kernel could be an incremental replacement for gamma memories, but it is unexplored so far. For any memory kernel function, c, which can be expressed as a weighted sum of gamma kernels,

$$c(t) = \sum_{j=0}^{\Omega} q_j c_{\alpha,j}(t). \tag{7.10}$$

Where q_j are the weighting coefficients, the resulting memory can be expressed in terms of the gamma memory state vectors:

$$x_t = \sum_{j=0}^{\Omega} q_j x_{\alpha,j}. \tag{7.11}$$

de Vries and Principe (1991) show that the gamma kernels form a basis set, meaning that, for an arbitrary kernel function c that decays exponentially to 0 as $\tau \to \infty$, there exists a value of Ω and a set of coefficients $\{q_j\}$ such that Eq. 7.10 holds. This result is easy to see for the case of delay kernels, but one can see from this simple case that the required computation amounts essentially to convolute a complex kernel with the input sequence at each time step is not of great practical utility.

On the other hand gamma memories combined with artificial stochastically spiking neurons model a biological more plausible onset behavior of the activation as the full depolarization of the neurons membrane to release a spike needs also some time. The combination of stochastically spiking neurons with a gamma memory might be of advantage in temporal sequence processing.

7.5 Contents of Short-Term Memory

Having described the *form* of the short-term memory, we should now discuss what the *content* looks like. Although the memory must hold information pertaining to the input sequence, it does not have to be a memory of the raw input sequence, as assumed previously. The memory encoding process can include an additional step in which the input sequence, $x(1) \ldots x(t)$, is transformed into a new representation $x'(1) \ldots x'(t)$, and it is this transformed representation that is stored in the memory. Thus, the x_is are defined in terms of x', not x. This transformation is a filtering of the input sequence. In this section we will follow the memory classification of M. Mozer [WG94].

The case considered in previous sections is an identity transformation,

$$x'(\tau) = x(\tau). \tag{7.12}$$

Memories utilizing this transformation will be called *input* or *I* memories. Models in the neural net literature make use of three other classes of transformation. First, there is a transformation of the input by a nonlinear vector function f,

$$x'(\tau) = f(x(\tau)). \tag{7.13}$$

This transformation results in a *transformed input* or *TI* memory. Generally f is the standard neural net activation function, which computes a weighted sum of the input elements and passes it through a sigmoidal nonlinearity:

$$f_w(v) = \frac{1}{1 + e^{-w \cdot v}}. \tag{7.14}$$

Second, the nonlinear transformation can be performed over not only the current input but also the current internal memory state:

$$x'(\tau) = f(x(\tau), x_1(\tau), \ldots, x_{\Omega(\tau)}). \tag{7.15}$$

This leads to a *transformed input state* or shorter *TIS* memory. Such memories can be implemented in a recurrent neural net architecture in which x_i and x' correspond to activities in two layers of hidden units.

Third, for *autopredictive* tasks in which the target output, $p(\tau)$, is a one-step prediction of the input; i.e.,

$$p(\tau) = x(\tau + 1), \tag{7.16}$$

one can consider an alternative content to the memory. Rather than holding the actual sequence value, the preceding prediction can be used instead:

$$x'(\tau) = p(\tau - 1). \tag{7.17}$$

Mozer named this transformation an *output* or *O* memory. Of course, *TO* and *TOS* memories can be constructed by analogy to the *TI* and *TIS* memories, suggesting a characterization of memory content along two subdimensions, one being the transformation applied and the other being whether the transformation is applied to the input or previous output. Since it does not make a great deal of sense to ignore input sequence information when it is available, I will include in the category of output memories those that combine input and output information, e.g., by taking a difference between input and output, $x(\tau+1)-p(\tau)$, or by using the input information when it is available, such as during training, and the output information otherwise.

7.6 Memory Adaptability

Mozer also discussed the possibility of implementing *adaptability* in short-term memories. In our scenario an adaptability of the neurons short-term memory is simply not necessary. Nonetheless the possibility should be discussed within a view short sentences.

All so far introduced memory definitions are static, due to their predefined and fixed feedback parameters. To render a *static* memory *adaptive*, it is necessary to make various parameters like $\{\alpha_i\}$,$\{\Omega_i\}$, and the w of Eq. (7.14) and Eq. (7.15) adaptable.

As stated before, if the parameters are fixed in advance, the memory can be called *static* as the memory state, $\{x_i(t)\}$, is a predetermined fixed function of the input sequence, $x(1)\ldots x(t)$. The task of an artificial neural network is to make the best predictions possible based on the given fixed representation of the input sequence history. Enabling an artificial neural network to control memory parameters, allows the memory representation being *adaptive*. By adjusting the memory parameters, the neural net delimits the capacity of the memory autonomously. This indirectly alters what aspects of the input data are taken into account for making predictions. The introduction of an adaptive memory is twofold. On the one hand side the system gets the benefit of a more accurate representation of the sequence, and on the other hand the system must also learn the characteristics of the memory.

In our search for an artificial neural network model, supplying the computational ability of the former defined orthogonal access memory, we decided to use an exponential trace memory being the instance which supplies an ultra short-term memory on the single neuron level. The next chapter shows several possibilities to introduce temporal memory structures to self-organizing maps. Two of the here introduced memory models are used in SOMs so far. A static memory is sufficient for the realization of the orthogonal access memory as Schill and Zetzsche proposed. As the static memory is sufficient, we will stick to this subset of memory models.

Chapter 8

Temporal Single Layer Networks

Before we can start defining our novel multi-layer model we first have to review some temporal enabled Kohonen style networks and much more important, we define some extended temporal processing self-organizing maps meeting our demands in the multi-layer application.

8.1 SARDNET

SARDNET [JM95] is a SOM variant for phoneme processing and representation. Its structure and learning algorithm is identical with standard SOM. The difference lies in the activation scheme of a single neuron. The best matching neuron –yet determined by its euclidian distance to the input vector– is activated by setting its activation level to 1.0 and it is also withdrawn from the pool of competitive neurons for the rest of the learning period. This causes one neuron to react to exact one phoneme transition. The activation of a single neuron is damped similar to the recurrent SOM neuron.

Learning is realized exactly like in SOM without any recurrent feedback of the neurons activation into the weight adaptation. In contrast to RSOM, SARDNET generates a pure state transition mapping whereas each transition has no relation to past transitions. One single neurons weight vector in RSOM is in contrast based on the combination of state transition of past events. Through the damping of the activation values SARDNET establishes a temporally ordered representation of the presented input, but each winning neuron is also taken out of the pool of competing neurons completely. By taking neurons out of the competition the generalization ability of the network is weakened. These points of critique motivated us to stick with RSOM derived neurons and to apply a nonlinearity to the RSOM neurons which we will introduce after a brief look on dynamic self-organizing maps.

8.2 Hierarchical Dynamic Self-Organizing Maps

The authors of the original paper on Dynamic Self-Organizing Maps (D-SOM) [PS96] developed their model to provide an intelligent system with the ability of processing patterns over time. Temporal pattern processing can be achieved by providing a short-term memory –like the ones introduced in the last chapter– so that different configurations of events can be persistent for a short period of time. In D-SOM, short-term memory is provided by leaky integrators forming a preprocessing layer in a self-organizing system. These leaky integrators are quite similar to the first temporal memory model, Tapped Delay-Line Memory. The described model exhibits a task, the authors of the original paper called compositionally.

> „. . . it has the ability to extract and construct progressively complex and structured associations in an hierarchical manner, starting with basic and primitive (temporal) elements . . . ".

An important feature of the D-SOM model is the use of temporal correlations to express dynamic bindings. Privitera and Shastri introduced the D-SOM in the context of a multi-layer processing model [PS96] for visual stimulus processing.

η	Learning rate
$P_i(t)$	Membrane potential
$u_i(\cdot)$	Input activation function
$x(t) \in \mathbb{R}^K$	Inputvector
$\mathrm{Act}_i(t)$	Set of the activated taps
τ_i	Time constant of neuron i
u_{ij}	Activation function of the j-th tap of the i-th neuron
$\mathbf{y}_i \in \mathbb{R}^Q$	Output weight second layer
$\omega \in \mathbb{R}^K$	Weight vector

Table 8.1: Symbol definitions DSOM

Weight adaption in D-SOM is learned by a classical Hebbian learning rule:

$$\Delta\omega_i(x) = \eta(x - \omega_i)u_i(x) \tag{8.1}$$

where x denotes the input vector, ω_i the weight, and u_i the activation function. D-SOM also provides a dimension reduction mechanism. Each element of the map is associated with a second weight vector $\mathbf{y}_i$. Whereas $\mathbf{y}_i \in \mathbb{R}^Q$ and $K > Q$. So each input vector x is associated to a $\mathbf{y}$ applying the following yet again Hebbian learning rule:

$$\Delta\mathbf{y}_i(x, \mathbf{y}) = \eta(\mathbf{y} - \mathbf{y}_i)u_i(x) \tag{8.2}$$

The activation function is given by:

$$u_i(x) = U_i(x) = \frac{G(\|x - \omega_i\|)}{\sum_j G(\|x - \omega_j\|)} \tag{8.3}$$

Whereas $G(\cdot)$ is a gaussian function with fixed variance of σ:

$$G(\|x - \omega\|) = \exp\left(\frac{\|x - \omega\|^2}{\sigma^2}\right) \tag{8.4}$$

In the D-SOM model, the activation of an element in the map is represented by the membrane potential $P_i(t)$ which in turn is a function of the input activation function $u_i(x(t))$ which measures the degree of match between the elements weight vector and the temporal input vector $x(t)$. $P_i(t)$ takes also into account the dynamic properties of the biological membrane of a cell which can be approximated by a generic RC-circiut.

$$\frac{\partial P_i}{\partial t} = -\frac{P_i(t)}{\alpha_i} + u_i(x(t)) \tag{8.5}$$

The membrane potential of an element is correlated with the temporal evolution of its input and the element can not only analyze a static input vector x, it can also perform a temporal integration of the function $u_i(x(t))$. If this temporal integration depends on the occurrence of certain spatio-temporal features/sub-events in the evolving temporal input, then the activation of the element indicates the occurrence of an external event composed of these sub-events. Thus one can view each element as a recognizer of a complex event.

Each element consists a set of sub-elements which are called taps Figure 8.1. Each of which represents different distinct points in the temporal evolution of the input vector. Specifically, if an element recognizes an event composed of m temporal features, it consists of m taps. If the input vector is k-dimensional, $\epsilon(t) \in \mathbb{R}^k$ then, each tap is characterized by a weight vector $\omega \in \mathbb{R}^k$. At each time instance, the input vector $x(t)$ is processed by all the taps and the membrane potential of the element is evaluated based on the tap outputs:

$$\frac{dP_i}{dt} = -\frac{P_i(t)}{\tau_i} + \sum_{j \in Act_i(t)} u_{ij}(x(t)) \tag{8.6}$$

where, $u_{ij}(x(t))$ is the activation function of the j-th taps of the i-th neuron and τ_i is the time constant of the neuron.

The term $Act_i(t)$ describes the set of active taps of the i-th neuron at a distinct time step. As soon as the activation function of a tap drops below a predefined threshold it is inhibited and excluded from the set of active taps. Different elements may have different activation strategies for taps.

The initial state of the taps can be active or inactive. In other cases, taps may become active one at a time, and hence respond to temporal features occurring in a specific order. This is realized by connecting the taps in a sequence as shown in Figure 8.1. Notice that taps are represented also by neurons, $tap_1, \ldots, tap_m$, that read a common input vector. Initially only the first tap is enabled. When an enabled tap receives a matching input signal, it crosses its threshold and fires, enables its successor tap, and at the same time inhibits itself thereby entering a refractory state.

The $n_i - th$ neuron integrates the tap output and maintains a continuous potential. This neuron eventually fires if all the taps have been activated by the temporal input. The time constant τ_i facilitates the delay limit between two sequential temporal features recognizing and the short term memory of the integrating neuron.

The learning of tap weights is done with a traditional Hebbian learning rule. Finally, the firing of a neuron is defined by:

$$x(t) = \begin{cases} \mathbf{y}_i & \text{if} \quad A_i(t) = 1 \quad \text{and} \quad P_i(t) = \min_{j:A_j(t)=1, P_j(t)>\Lambda(t)} P_j(t) \\ 0 & \text{if} \quad \forall_{iA_i(t)} = 0 \end{cases} \tag{8.7}$$

$A_i(t)$ is the total inhibition of the element which equals 1 this holds only if all the taps of the element have been matched during the analysis of the input sequence and 0 otherwise. $\mathbf{y}_i$ is the output weight vector of the referred element. $\Lambda(t)$ is a common threshold function. Though having some of the

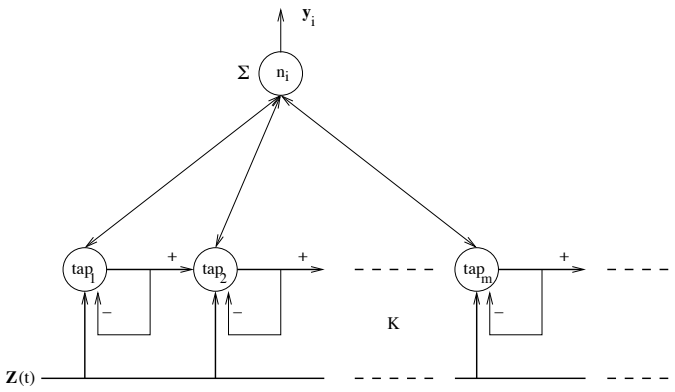

Figure 8.1: Structure of a partitioned leaky integrator neuron

properties we need to process spatio-temporal information, the model fits not all demands. The temporal aspect and an orthogonal access as described in [SZ95] are not taken into account. This model also does not meet the demand of a prediction functionality which is realized in the human visual processing system and very important for motion processing in natural environments. Nevertheless this model gives some good design ideas for the later introduced model.

8.3 Recurrent Self-Organizing Maps

There are several ways to integrate time into a SOM-like network. One is the introduction of a shift register like structure discussed in the section above, another in our opinion the most biological way, is the damped feedback of the neurons output to its input as illustrated in Figure 8.2. We focus on the recurrent Self-Organizing Map (RSOM) like discussed by Koskela [KVHK98] because of the similarity to the organization in the visual cortex of the human brain.

$x(t)$	Input vector at time step
$y_i(t)$	Output of neuron i at step t
$y_b(t)$	best matching neuron
$w_i(t)$	weight vector of neuron i at step t
α	damping factor
γ	learning rate
N_{ib}	neighborhood function
θ_s	Significance threshold

Table 8.2: Symbol definitions RSOM

In contrast to the original intention behind the RSOM algorithm we are interested in the *activation pattern* a sequence of input values is evoking on a RSOM layer. This pattern is the information representation which incompanies spatial and temporal knowledge that can be further processed in a hierarchically higher processing level.

To implement time in the SOM model introduced earlier we consider the input values $x(t)$ being a sequence of values dependent on time t and add a feedback loop to each SOM neuron. This loop feeds after a delay of one time-step the activation of each neuron back to its input while being damped by the factor α, $0 < \alpha \leq 1$. Figure 8.2 shows the signal flow diagram of a single RSOM neuron. The output of a recurrent neuron is a combination of

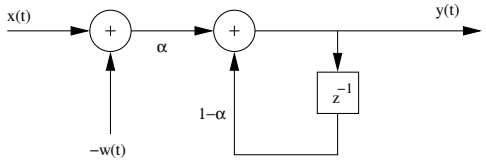

Figure 8.2: Signal flow diagram of a RSOM neuron

the damped output of the last time step, with the current output:

$$y_i(t) = (1-\alpha)y_i(t-1) + \alpha(x(t) - w_i(t)) \tag{8.8}$$

Thus we do not calculate the distance between input vector and weight vector, the best matching neuron b at time step t is the output vector with the smallest norm:

$$y_b(t) = \min_i \{||y_i(t)||_p\} \tag{8.9}$$

The temporal behavior of a neuron is determined by its response $h(k)$ for all $k \geq 0$ to one single input of maximal size, at time 0[1]. As only the recurrent feedback Figure 8.2 is relevant for the temporal behavior of the neuron, $w_i(t)$ is neglected. A normalized input is applied by setting the input value at time step $t = 0$:$x(0) = 1$. At every other time steps $t \neq 0$ the input is $x(t) = 0$. The resulting response $h(k)$ is the so called system response

$$\begin{aligned}
t = 0 : h(0) = &\quad (1-\alpha)\cdot 0 + \alpha \cdot 1 = &\quad \alpha \\
t = 1 : h(1) = &\quad (1-\alpha)\cdot \alpha + \alpha \cdot 0 = &\quad (1-\alpha)\cdot \alpha \\
t = 2 : h(2) = &\quad (1-\alpha)^2\cdot \alpha + \alpha \cdot 0 = &\quad (1-\alpha)^2\cdot \alpha \\
t = 3 : h(3) = &\quad (1-\alpha)^3\cdot \alpha + \alpha \cdot 0 = &\quad (1-\alpha)^3\cdot \alpha \\
&\quad \vdots &\quad \vdots &\quad \vdots \\
t = k : h(k) = &&&\quad \alpha(1-\alpha)^k
\end{aligned} \tag{8.10}$$

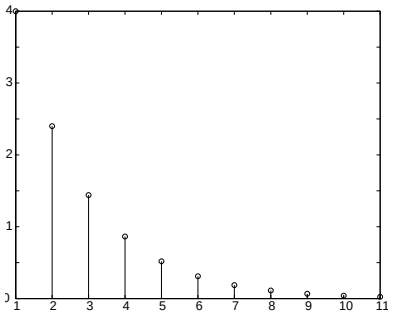

Figure 8.3: System response $\uparrow \alpha = 0.4 : \rightarrow t$

As we can see, a recurrent neuron reacts with a discretized exponential decrease of its output to being activated Figure 8.3. The decreasing output values of all neurons are molding a temporally ordered output pattern, as sketched in Figure 8.4. Through this pattern we can access the spatial and the temporal properties simultaneously what we call an orthogonal input access. The demand of an orthogonal memory access was one key topic of the dynamic spatio-temporal memory model of K. Schill and C. Zetzsche [SZ95] we used as a foundation to develop our model. The weight adaptation

[1]For signal theory background see [PRLN92].

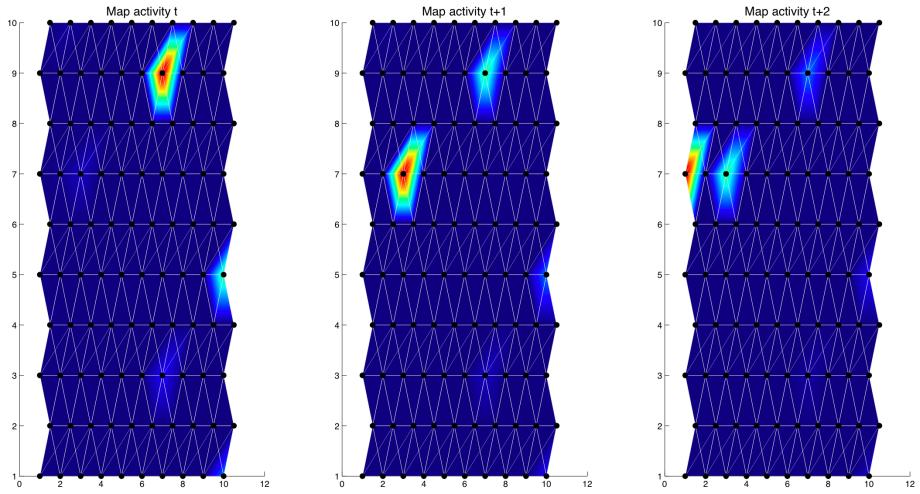

Figure 8.4: Activation trace of a single STRSOM Layer

is extended by replacing the simple distance measure by the recurrent output calculation in Eq. (8.8)

$$w_i(t+1) \quad = \quad w_i(t) + \gamma(t) N_{ib}(t) y_i(t) \tag{8.11}$$

To enable the initial SOM network for temporal processing the weight adaptation scheme also has to be changed to address temporal processing. In Eq. (8.11) the common euclidian distance is replaced by $y_i(t)$ which holds the recurrently damped activation of past states of each neuron.

Training a non-recurrent SOM is done by random sampling of an input vector out of the training set and moving the best matching neuron and its spatial neighboring neurons, selected with an neighborhood kernel, in direction to the supplied input vector. Drawing the input vectors randomly out of the training set prevents the system from poor adaptation, a twisted net would for example lead to low generalization ability of the trained network. As mentioned above, using a RSOM model we can process temporal information. To do so, the ordering of the input values is of crucial importance and holds a very important part of the information we want to process. A simple random draw of input values to prevent poor adaptation is therefore not possible any more. We will now introduce our novel data handling method for the weight adaptation.

By dividing the input data into subsets –which we call subsequences– it is possible to realize good training results and to preserve the temporal structure of the input data without having problems with a bad generalization

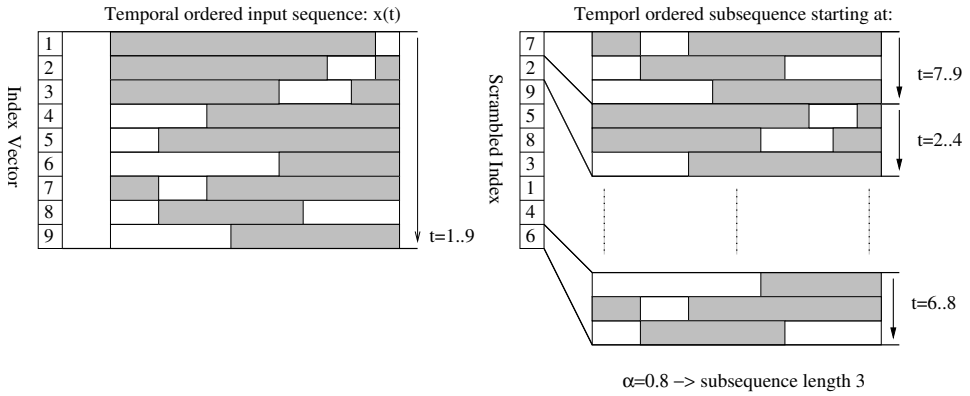

Figure 8.5: Subsequence Learning

performance of the network. We use the system response and a predefined threshold to determine the length of a subsequence.

Definition: *Let us state that a neuron with a remaining activation level of* 5% *of its original activation holds no relevant information for the processing any more. We will refer to the number of time steps necessary for the activation to drop below* 5% *as the* event horizon *of a neuron. The threshold of* 5% *is denominated with* θ_s*.*

We use this threshold θ_s to calculate the subsequence length by applying the system response of a neuron. Within each of the subsequences the temporal ordering of the input vectors is assured. To ensure that all input values are processed, a permutation of the index vector enumerating the input vectors is generated. The permuted index is defining the entry point of each subsequences in a pseudo random fashion as illustrated in Figure 8.5. This scheme of choosing the input vector prevents the system from bad adaptation and also preserves the temporal context of the input. It is sufficient to use this scheme in the first ten percent of training cycles, as the unfolding of the network is done in the very early training phase.
The RSOM algorithm described in [KVHK98] was originally introduced for the purpose of time series prediction. This kind of prediction was realized by extrapolating the direction in which the last winning neuron was pointing to. This behavior is not necessary for our intention. We are using the decayed feedback of the neurons to let the network form a temporally ordered activation pattern of past winning neurons. The next chapter introduces our multi-layer model based on RSOM networks, but we will introduce some novel single layer SOM networks first.

8.4 Strict Temporally Ordered Map

Taking the properties of the RSOM algorithm into account, it becomes obvious that a temporally ordered activation pattern is not assured at every processing time step. The reason for this lies in the way the activation level of each neuron is calculated. Basing the activation calculation mainly on the euclidian distance it gets possible, that the stimulus' distance to its best matching neurons weight vector is rather big and the last winner neurons activation could have still a higher activation level then the winner neuron of the current time step. The error is not only in the temporal domain it is very likely also spatial, because the direction of last winning neurons weight vector is not necessarily similar to the actual winning neurons weight vector.

To realize a strict temporally ordered map (STORM) it is necessary to calculate a dendritic excitation, based on the distance of the weight vector to the input vector and an axonal part similar to the spike of a biological neuron. If the excitation of the neuron is higher than a predefined threshold and the neuron is not inhibited due to a former activation, it is generating a standard spike. The axonal part is identical to the RSOM neuron in Figure 8.2. Obviously this assures a strict temporally ordering of the activation pattern in the map. The best matching neuron y_b to the presented input is determined again by the minimum vector length found by applying Eq. (9.2). We call the best matching unit the strongest excited neuron, as the input matches best to the neurons weight.

$$y_b = \min(|y_i|) \tag{8.12}$$

After finding the strongest excited neurons and generating a sorted index vector of the neurons excitation, a standardized activation value is assigned to the first non-refractory unit. All other activation values are damped by the factor $(1 - \alpha)$. The activation value is determined through:

$$\begin{aligned} z_b(t) &= 1.0 \\ z_{i \neq b}(t) &= z_{i \neq b}(t-1)(1-\alpha) \end{aligned} \tag{8.13}$$

By applying an unified activation to the first non refractory neuron, we establish a nonlinear projection from the input to the output of the neuron. Calculating the activation in this way ensures a strict temporally ordering of the *firing* state of the neurons. The result of this scheme is an activation pattern of all neurons whereas the level of activation also codes it temporal occurrence in the past.

The period within a neuron is stated refractory is determined with the system response and the significance threshold θ_s.

$$k_\theta = min_k | \theta_s \geq \alpha(1-\alpha)^k \tag{8.14}$$

Creating a strict temporal ordered mapping is crucial for the further temporal processing we want to realize in the later introduced multi-layer network.

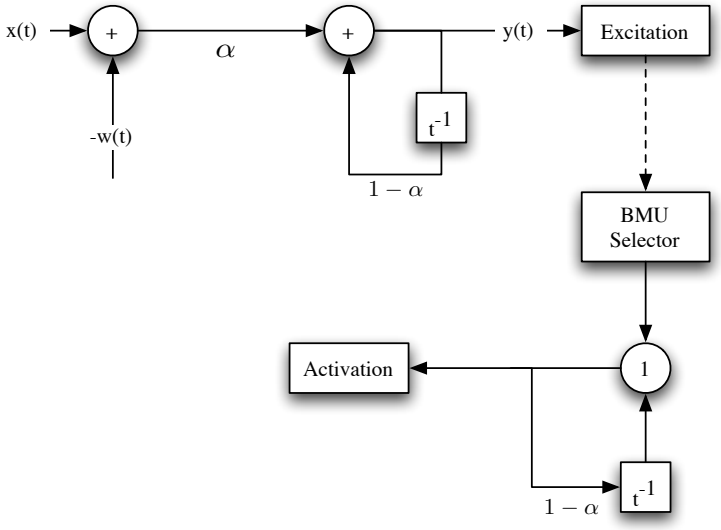

Figure 8.6: STORM Neuron with separated excitation and activation

8.5 SARDSTORM

To investigate the performance of our new STORM algorithm we defined another flavor of STORM with respect to SARDNET. We believe that the recurrent weight adaption is of crucial importance for sequence processing. SARDNET is a network definition which has non recurrent weight adaption combined with an exponential decaying activation of the neurons. Only the fact of removing an once activated neuron from the pool of competing neurons does not fit our attempt of a comparison. We therefore define SARDSTORM a model, which is a mixture of SARDNET and STORM.

The weight adaptation is identical to the definition of the standard SOM network. A once selected best matching neuron gets an standardized activation value which is decreasing exponentially. Within an predefined interval the neuron is stated as refractory and therefore withdrawn from the competition unless the activation falls the significance threshold.

Algorithm 2 Common training loop

```
γ = 0.5;
σ² = 2.0;
init(Layers);
random_init(w_i(t)^n);
permuted_index = generate_perm_index(number_of_trainsamples);
for i = 0 to trainiterations do
  for t = 0 to number_trainset_entries do
    if STORM then
      STORM_train(permuted_index(t));
    end if
    if SARDSTORM then
      SARDSTORM_train(permuted_index(t));
    end if
    if stRSOM then
      stRSOM_train(permuted_index(t));
    end if
  end for
end for
```

Algorithm 4 SARDSTORM_train(index)

```
k_s = calculate_subsequencelength(α, θ_s);
for t = 0 to k_θ do
  for i = 0 to numberofneurons do
    y_i(t) = (1 − α)y_i(t − 1) + α(x(t) − w_i)
    activations_i = (1 − α) · activations {damp old activations}
  end for
  bmu_index=sort(y,ascend)
  repeat
    b=bmu_index(i)
    i++
  until activations(y_{bmu_index(i)}) ≤ θ_s
  set winning neuron refractory
  for i = 0 to numberofneurons do
    w_i(t + 1) = w_i(t) + γ(t)N_ib(t)(x(t) − w_i) {standard weight adaption}
  end for
  activations(y_b(t)) = 1.0
end for
```

We use this model only to show the difference between the processing of single state transitions and multiple state transitions in the experimental chapter.

Algorithm 3 STORM_train(index)

k_θ = calculate_subsequencelength(α, θ_s);
for $t = 0$ to k_s **do**
  for $i = 0$ to numberofneurons **do**
    $y_i(t) = (1-\alpha)y_i(t-1) + \alpha(x(t) - w_i)$
    $activations_i = (1-\alpha) \cdot activations$ {damping of old activations}
  end for
  bmu_index=sort(y,ascend)
  repeat
    b=bmu_index(i)
    $i++$
  until activations($y_{bmu\_index(i)}$) $\leq \theta_s$
  set winning neuron refractory
  for $i = 0$ to numberofneurons **do**
    $w_i(t+1) = w_i(t) + \gamma(t)N_{ib}(t)y_i(t)$ {recurrent weight adaption}
  end for
  activations($y_b(t)$) = 1.0
end for

8.6 Stochastic Spiking RSOM

In this section we will do a small step towards a slightly more biological plausible neuron definition. STORM neurons already have some extension to make the activation reaction more like in real neurons: the refractory period and the nonlinear separation of the excitation and the activation of each neuron. To complete the changes to the neurons we introduce stochastic spiking neurons and a neighborhood function which is basically gaussian, but used for the adaptation of the spike probability.

To integrate a stochastic spiking activation behavior into a RSOM neuron we keep the euclidian distance measure between input data and weight vector to determine the dendritic potential, but replace the activation calculus by a probability distribution footing on the euclidean input/output distance.

Θ	Threshold function
t_s	Time at which neuron spiked
p	Probability of spike
F_β	Fermi function

Table 8.3: Additional Symbols for Stochastic Spiking RSOM

$$y_i(t+1) = (1-\alpha)\cdot y_i(t) + \alpha \cdot |x(t) - w_i(t)| \tag{8.15}$$

$$z_i(t) = \begin{cases} 1: & \text{with prob. } p = f(y_i(t) - \Theta(t - t_s)) \\ 0: & \text{other} \end{cases} \tag{8.16}$$

With $f : \mathbb{R} \rightarrow [0,1]$ defining a monotone, growing function, e.g. the fermi function $F_\beta(y) = 1/(1+\exp(-\beta y))$ with $\beta > 0$. Θ is defined by a monotonically decreasing threshold function with $\Theta : \mathbb{R}^+ \rightarrow \mathbb{R}^+$. The probability p of a new spike appearing is proportional to the length of a simulation step δt. For constant Θ the spiking neuron degenerates to a simple threshold neuron. By choosing the heaviside function for f the stochastic neuron becomes a deterministic spiking neuron.

The weight adaption is identical to the previous definition Eq. (8.8), but the interpretation of the neighborhood function is slightly different. Normally the gaussian neighborhood is altering the width of how much the weight vectors in the surrounding of the winner neuron are adapted towards the input datum. Here the activation is a transformed value based on the distance, but expressed by a probability measure. By decreasing the distance between input and weight the probability of a released spike increases. The gaussian neighborhood function is shaping the spike release probability around the winning neuron.

Further investigations on this model are still necessary an will be addressed in future work.

Chapter 9

Multi-Layer Spatio-Temporal Information Processing Models

To link the results of the psychophysical experiments describing a macroscopic view of the systems capability, and the neuro-anatomical findings which deliver a microscopic view of the motion perception abilities of the visual system with artificial neural networks based on Kohonens self-organizing maps [Koh97], we introduce a novel approach of a multi-layer neural network in this chapter. The view on the model introduced in this chapter is related to a view on the visual system, we intend to call mesoskopic. We will provide a model that has the key processing capabilities of the biological system for a reasonable cost of computing, we will not define the network structure exactly like the so far examined natural visual system of the macaque brain.

We start with some established models we examined in the process of model finding. All of the discussed established models fulfill our demands only partially. The capabilities which are not met are addressed and discussed in place. All models –for example– don't meet the demand of an orthogonal access to space and time to name only the major point of critique. Due to the fact, that within this thesis we intend to develop a biologically inspired model we will stick to the artificial neural networks used in our model therefore discarding Dynamic Bayesian Networks, Hidden Markov Modells or Kalman/Particle Filter ensembles.

We restrict ourself to a short statement on statistical models to allow to point out some similarities between the neural network models we introduce and statistical models. Looking at self-organizing maps from the statistical point of view you might argue that a SOM layer neurons fit to a provided input could be modeled by a conditioned probability. So to say the neuron gets activated if the probability that the input is similar to the value a neuron is representing. Using such a scheme would lead to a model based an normal distributions. Making a step towards recurrent normal distributions we get to Kalman filters or in a discretized environment to particle filters. Also

dynamic bayesian networks could give a proper representation of state space formulations. We discarded those models not only because of the fact, that they are not biological plausible. They also lack key properties we want to realize in our model. Some issues are the orthogonal access to space and time at each time step, prediction capability which goes further than simple extrapolation, to allow a context switch within prediction phase.

We will now continue with the introduction of some established multi-layer Kohonen based neural network models to provide some background before we continue with our new models Spatio-Temporal RSOM and spatio-temporal STORM.

9.1 Recurrent Kohonen Self-Organization in Natural Language Processing

This two-layer model based on self-organizing maps, was introduced by J.C. Scholtes [Sch91]. The central point of this model is the system's ability to process state transitions within the higher more abstract processing layer. This ability is realized by the feedback of the output of the most abstract layer to the input of the first processing layer. The model was intended for

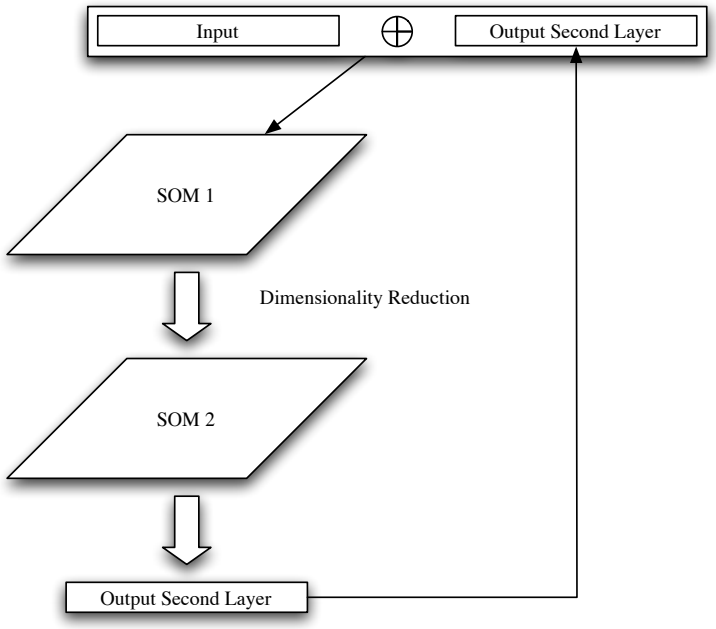

Figure 9.1: Recurrent Kohonen Self-Organization in Natural Language Processing

natural language processing. It was one early multi-layered approaches using Self-Organizing maps for context aware sequential data processing. Compared to the demands we have for a motion processing system "Recurrent Kohonen Self-Organization in Natural Language Processing" lacks prediction capability and even more important, it does not provide any orthogonal access memory or even any short-term memory. The dimensionality reduction realized with a predefined heuristic, between the two processing layers also prevents an orthogonal access to space and time for information derived from the first processing layer also in the higher processing levels.

9.2 Adaptive Resonance Theory

Although generally classified as a neural network, Adaptive Resonance Theory (ART) is sufficiently eclectic that it warrants a separate discussion. The original theory of ART networks was proposed by Stephen Grossberg [Gro86], and implemented by Carpenter and Grossberg [CG87].

An ART network performs an unsupervised batch clustering of input data. Given a set of input patterns, an ART network attempts to separate the data into clusters of similar patterns. An ART network consists of two processing layers –nodes–, which form an iterative feedback loop. The first layer acts as short-term memory; the second layer represents a cluster in the set of input patterns and contains the node prototype representing the centre of the cluster.

The number of nodes in the second layer is allowed to change as required, in order to represent the input patterns best. The creation of nodes in the second layer is controlled by a parameter ρ known as the *vigilance* of the network. This parameter determines the granularity of the clusters represented by the second map, by acting as a threshold on the similarity between clusters represented by nodes.

The *resonance* in ART occurs in the iterative feedback loop between the two processing layers, when output between the two layers are within some threshold of similarity –that is, the output vector $F_1 \rightarrow F_2$ is similar to the output vector $F_2 \rightarrow F_1$–. The selection of a winning node in an ART network utilizes this resonance. Presentation of an input pattern activates the nodes in the first layer, which propagates the input vector through a set of fully connected weights to the second layer. This activates the nodes in the second layer, and the node with the highest inputs initiates a feedback loop with the first layer. This alters the values of the nodes in the first layer, which in turn alters the second layer, and son on. This process terminates when a node in the second layer places the network in a resonant state; this node is then labeled as the winning node. If no node can be found to place the network in a resonant state, the networks adds a new node to the network, and declares this new node to be the winner.

There are a large number of variations based on the ART concept. For example:

- ART2 allows the clustering of analogue input vectors
- ART3 adds bias to the search process
- ARTMAP uses a pair of ART networks to create a network capable of supervised learning

To name only a small subset of all ART like models. Each of these networks are based on the same basic principles defined by the original ART network. As a result, ART based architectures, regardless of the specific implementation, tend to suffer from some common problems. As a result of the Winner-Take-All competitive nature of ART networks, they are prone to local minima in the search space. They are also prone to over-fitting, especially in the presence of noisy data. They are also potentially sensitive to the order of presentation of input vectors, which is in fact the biggest disadvantage in temporal processing.

Just as with neural networks, ART networks can be easily applied to continuous learning problems. The naïve approach of continuously presenting new training data will result in a network which continuously adds new information to the network. However, just as with standard neural networks, ART networks are prone to plasticity problems. Over time, the network will tend to drift away from concept representations which are not reinforced through training.

This plasticity issue has been addressed in variants of ART. One of these variants IFOSART, maintains the learning rate of the network as a whole above a given threshold, while allowing the learning rate of individual neurons decay to a stable minimum. The decay in individual neuron learning rate, new nodes can be added to the system as required.

ART architectures have proven themselves to be good unsupervised (in the case of ARTMAP of course supervised) classifiers for general learning problems. They are also well suited to continuous problems, and have been successfully used for this purpose in a number of situations, but they are everything else than easy to handle. Many papers in natural language processing are mainly on finding proper parameters for the used ART networks. The biggest disadvantage of ART, at least in our opinion, is the missing topology conservation a network needs to generate a retinotopic mapping of the input data into the state space.

9.3 Learning and Processing Sequences (LAPS)

Adaptive Behavior Cognition (ABC) introduced in the PhD. thesis of G. Briscoe [Bri97] is a quite general attempt of an artificial model that is able

to simulate a part of the cognition apparatus of the human brain. He created a neural network model that combined three schools into one cognition model: Behaviorism, Cognitivism and Connectionism. ABC in its complete realization is more a proof of concept than an applicable model, it combines visual/auditive perception with motor control by several instances of self-organizing maps, connected in a proper way.

Only the functional core of ABC, Learning and Processing Sequences (LAPS), was analyzed through simulations in full scale and more deeply by the author, and also integrated into an automated mapping support system. It is sufficient to discuss LAPS as this part of ABC describes the functional core, lacking only multi-modal interactions.

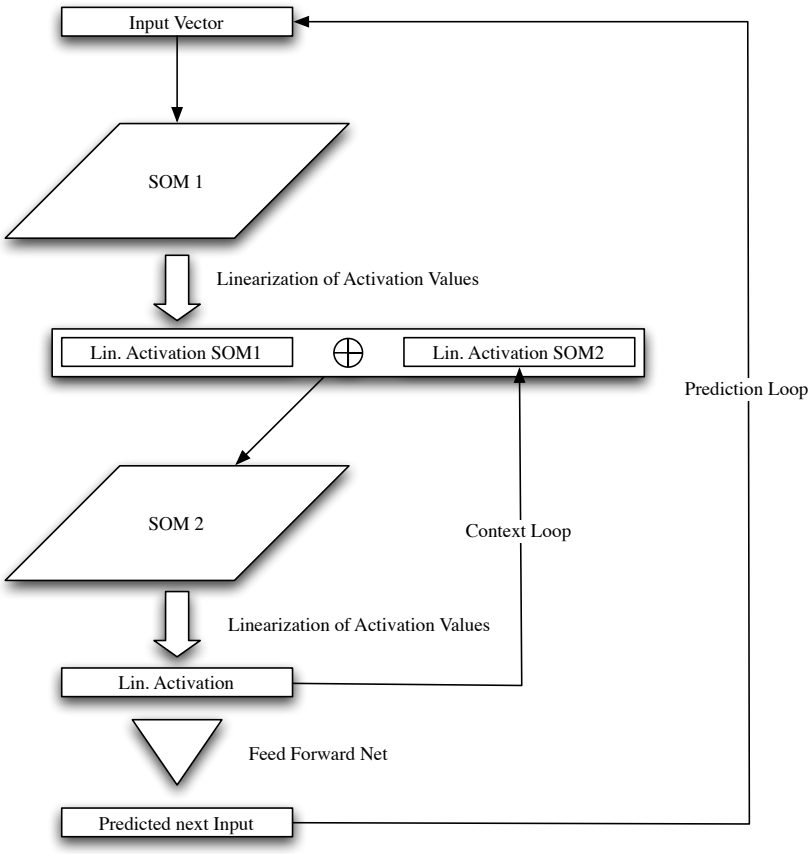

Figure 9.2: Learning and Processing Sequences

LAPS itself is as mentioned already a multi-layer model, built out of two SOM layers and an additional feed-forward network, which realizes prediction capability Figure 9.2. As this model uses no recurrent neurons like RSOM or a shift register like structure in D-SOM, one single processing layer

has no short-term memory. The temporal memory is realized by a textual concatenation of the state description of the first SOM layer with the state description of the second layer. One can say that the input presented to the second layer consists of something like state transitions.

The state description of a single SOM layer $\mathbf{y}_s(t)$ with n neurons is derived by the enumeration of all neurons scalar activation values $y_i(t)$ to form a single vector. We will refer to this kind of transformation as *linearization* later on.

$$\mathbf{y}_s(t) = y_1(t), y_2(t), y_3(t), \ldots y_n(t) \tag{9.1}$$

By concatenating the state description of the first and the second layer $\mathbf{y}_s^{<1>}(t-1) \oplus \mathbf{y}_s^{<2>}(t)$ the input to the second processing SOM is generated on the fly. As easily can be seen, the second processing layer is generating a state transition mapping from a compound of earlier stages to the momentary activation description of the first processing layer. In this case the first layer acts like a preprocessing filtering stage to transform the provided input value to an activation mapping, which again can be described as a low-pass filtering of the input signal.

With its feed-forward network LAPS realizes a direct input-value prediction. The additional feed-forward network connected to the second processing layer learns an association of the current state description of the second SOM-layer to the next upcoming input value which will be presented to the first processing layer.

LAPS predicts what phenomena is likely to occur next and not how the state description of the first processing layer will look like in the next future time step. LAPS is able to generate sequences if provided with a view starting input values.

Critics to address our demands arise from the fact that the system has, like Scholtes' network, no short-term memory with any orthogonal access to space and time in on single processing layer stage. Prediction capability is implemented but in a not very biological way.

9.4 Spatio-Temporal RSOM

So far all introduced models addressed either spatial or temporal information processing capabilities.

LAPS learns single state-like representations in the first processing layer, and state transitions within the second layer. The first layer has as stated above no short-term memory on the single neuron level and is thus not capable to store contextual information over several time steps in the higher processing layer. This lack of processing power brings up several ideas on extending the so far introduced models. The next step is to introduce short-term memory capabilities to each layer, what we have done in the spatio-temporal

RSOM model. Based on LAPS and the RSOM algorithm we developed a motion-perception model that is able to classify and predict spatio-temporal sequence information and at the same time to provide an orthogonal access memory demanded by Schill and Zetzsche [SZ95]. The boundary conditions of the model and several design issues were constrained by our main aim, the simulation of the behavior of the visual-motion processing area (area MT or V5) of the human brain other design issues where provided by the results of the psychophysical experiments discussed in the related chapter earlier. These investigations lead to some publications on spatio-temporal processing structures [BKRB00], [RSB+03], and [BB03].

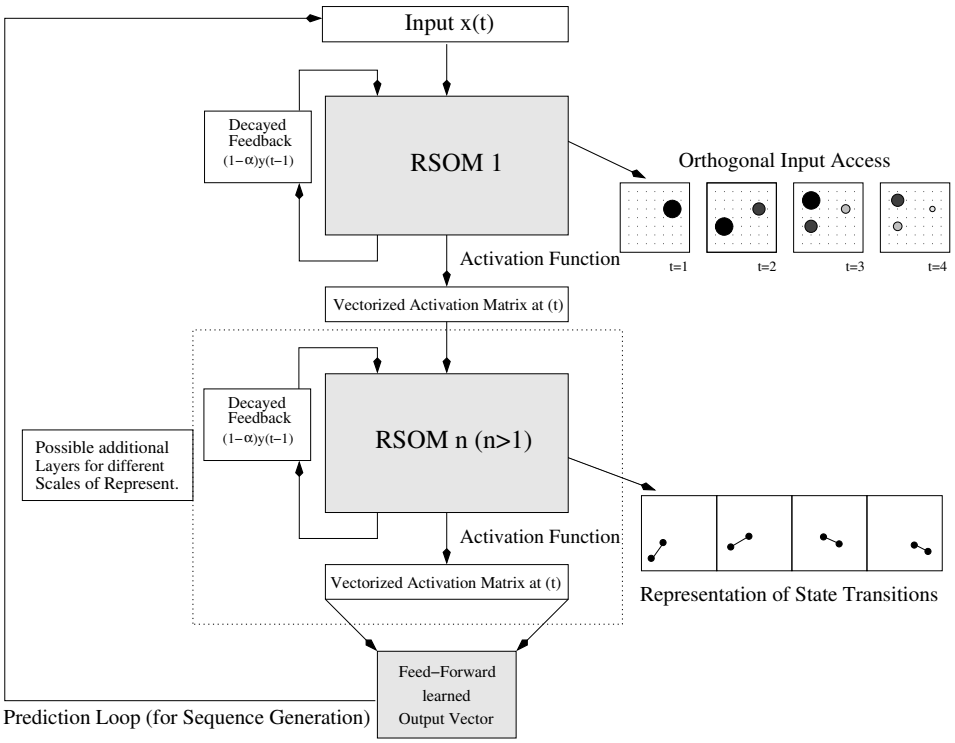

Figure 9.3: Spatio-Temporal RSOM

The complete structure of the model is shown in Figure 9.3. The RSOM layers are handled by the standard algorithm described above, apart from the output of the neurons. Since we try to base the model on biological findings, the maximum activation normalized by the dimension d of the weight vectors is calculated to simulate a biological neurons' output:

$$y_i(t) \quad = \quad (1-\alpha)y_i(t-1) + \alpha \cdot \exp\left(-\frac{||x(t)-w_i||}{\sigma^2 \cdot d}\right) \qquad (9.2)$$

The best matching neuron y_b is determined by the maximum activation:

$$y_b = \max(y_i) \tag{9.3}$$

To address multi-layer processing, information has to be passed from one processing layer to the next one. The input for the higher order network layer $z(t)$ is generated by combining all calculated activations ($y_i(t)$) of the neurons of the first map to one long vector through textual concatenation.

$$z(t) = y_1(t) \oplus y_2(t) \oplus y_3(t) \oplus \ldots \oplus y_m(t) \tag{9.4}$$

This successive generative process is bound by the *system response* of the RSOM neurons, which is controlled by the feedback parameter α. The closer α gets to 0, the longer the time interval the system "delays" the information propagation to the higher order processing layer. By setting the delay to several time steps, RSOM generates an internal activation pattern represented by the activation states of past innervated neurons.

Let us propose, that if the activation value of a past winning neuron falls below five percent of the original activation value, the neuron has no influence to overall processing any more, so to say no significant information. The time span between the initial activation of a neuron and its activation falling below this threshold delimits inherently the number of input vectors composing a subsequence and is tuned by the parameter α Eq. (9.2).

The time interval between two generated input-vectors for an auxiliary processing layer is delimited indirectly by the system response Eq. (8.10) of each neuron. As we want to preserve the temporal structure –the context– of the input-data we have to choose the time span shorter than the subsequence length. This ensures an overlapping of the subsequences and therefore the complete sequence representation stays well-defined. The above introduced elements of the model limits its capability to classification tasks. We extend the model by adding an alternate network structure responsible for prediction. This prediction extension allows two forecasting mechanisms: one-step prediction and dreaming/open prediction. The prediction structure of the first of the multi-layer models we introduce in our work is realized simply by a feed-forward net. This prediction network is trained with the standard back-propagation [Hay94] algorithm to learn an association of the current activation pattern of the second RSOM layer $z(t)$ Eq. (9.4) with the next upcoming input vector $x(t+1)$ of the first RSOM layer.

The RSOM is an unsupervised trained network whereas the feed-forward net connected to the second processing RSOM layer is self-trained through the system states of second layer. Through the fact that the training input of the feed-forward net is provided *on the fly* by a generative process of the preceding layers, we name the training scheme of the feed-forward network

Algorithm 5 Spatio-Temporal RSOM training algorithm

```
Establish initial value of γ, σ²
Initialize Layer1, Layer2
Init neurons feed-forward net
input dim = number of neurons layer2 output dim=input dim Layer 1
Random initialization of neuron weights all layers
for i = 0 to trainiteration do
  for j = 0 to number_trainset_entries do
    RSOM_train Layer1
  end for
end for
generate_input for Layer2 respecting event horizon Layer1
for i = 0 to trainiteration_layer2 do
  for j = 0 to number_generated_entries do
    RSOM_train Layer1
  end for
end for
generate input for feed forward net
for i = 0 to trainiteration_feedforward do
  for j = 0 to number_generated_entries_prediction do
    FeedForward_train
  end for
end for
```

semi-supervised. By the introduction of additional RSOM layers, it is possible to generate representations for more coarse time scales. As a result we can represent motion primitives, like “left-turn”, “u-turn”, “right-turn” or whole motion plans by the activation of one neuron in the highest processing layer.

9.5 Spatio-Temporal STORM

Not very pleased with the fact that the feed-forward network, used for prediction in the model introduced in the last chapter, is not very likely the realization of the prediction mechanism in a real brain and also not being very efficient in terms of learning and computing time, motivates the development of an alternate new prediction mechanism [Bai05].

We argue for an indexing structure which stores recurrent connections between neurons and the gating mechanism for retroinjecting connections, which are responsible for a priming and threshold decreasing of the most likely to be active neurons at the next upcoming time step. The *biological economy principle* states that unnecessary use of energy doesn’t make any sense in an biological organism. Since the brain consumes a big bunch of

the available energy, it is obviously of great importance to save energy by controlling the activations of the neurons in a reasonable way. The priming of the next most likely active neurons enables the brain to react quicker to an upcoming percept. This priming can also be interpreted as a kind of prediction mechanism. If an anticipated percept is missing –e.g. a moving object occluded by an obstacle, as an example of the visual processing– the brain is capable to estimate those missing stimuli to complete an imagined trajectory.

In the Spatio-Temporal STORM we replaced each single processing layer of stRSOM with a STORM layer. Using our new defined structure allows us to assure that the temporal ordering of the activation pattern is preserved at every time step and the demand of an orthogonal access to space and time in each processing layer is fulfilled.

The prediction loop now is realized by a hetero neural associative memory [PS95] for binary patterns –in literature often also called Content Addressable Memory/CAM–. This prediction memory stores the association of the firing state of the maps neurons and associates the derived activation state vector with the activation state of the next expected firing neuron. As mentioned above, this assembly should be seen as an indexing structure which stores the activated feedback connections, and not as a sole network structure.

Figure 10.1 shows the smallest functional part of stSTORM. This single processing layer already has the processing power of the earlier introduced LAPS assembly and it is additionally more efficient to train and more efficient in memory consumption.

The activation status of a single neuron is not defined by a singular scalar value $(0, 1)$. By choosing k_θ-tuples of binary *style* values it is possible to represent the strength of activation potentials. For example $< 1, 0, 0 >$ defines the most active neuron at the current time step <0,1,0> refers to the former active neuron which has already been damped. We don't use the classical logarithmic number representation like in the binary number system, it is more a simplified state description of the referred neuron. We get the complete state description vector by filling the k_θ-tuples of all non active neurons simply with zeros.

Through this kind of *binary* activation representation, it gets possible to store the activation pattern in a binary hetero associative memory. As stated earlier the associative memory is a very efficient way to store information in a robust manner.

To ensure that the associative memory gets loaded sparse enough, a proper representation has to be chosen. Once again we get back to what we earlier called the *event-horizon* of a single neuron. This horizon delimits the length of the tuple which defines the representation of the activation state of the addressed neuron. Now let us define the possible activation levels for $k_\theta = 3$:

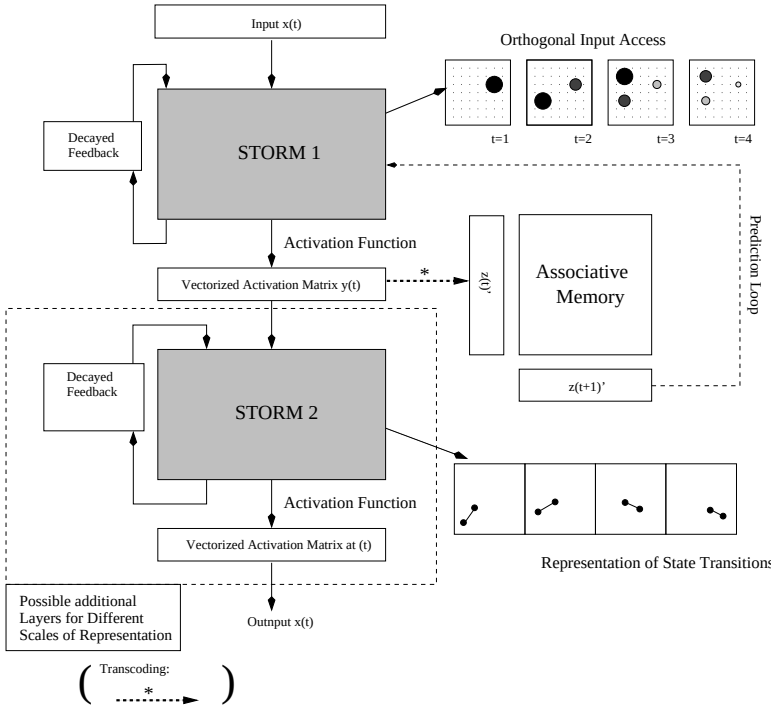

Figure 9.4: Spatio-Temporal STORM. Memory representation layer 1: activation patches with orthogonal access. Layer 2: combined activation patches with orthogonal access.

The position of the 1 in a k_θ-tuple denominates the activation strength of the referred neuron. A damping in this representation is done quite easy by shifting the token (1) every time step by one position to the right. If the token reaches the right most position, it gets deleted the next time step.

<1,0,0>	first time step
<0,1,0>	second time step
<0,0,1>	third time step
<0,0,0>	fourth time step

Table 9.1: Temporal development of the transcoded activation of a single neuron

So far we have only defined the input representation for the associative memory. The output representation depends on the purpose. We distinguish between two cases. The first case is used for the one step prediction. For this purpose we use a vector of the length n –the number of neuron– in which a single token represented by a 1 addresses the in the next time step activated

neuron. All other values in the output vector are simply set to 0. With this association we realize an one step prediction with a single retrieval operation.

Algorithm 6 Spatio-Temporal STORM training algorithm

```
Establish initial value of γ, σ²
Initialize Layer1, Layer2
Initialize Associative Memories Layer 1, 2, 3, Blackboard
Random initialization of neuron weights all layers
for i = 0 to trainiteration do
  for j = 0 to number_trainset_entries do
    STORM_train Layer1
  end for
end for
generate_input for Layer2
for i = 0 to trainiteration2 do
  for j = 0 to number_trainset_entries2 do
    STORM_train(Layer2)
  end for
end for
generate_input for Layer3
for i = 0 to trainiteration3 do
  for j = 0 to number_trainset_entries3 do
    STORM_train(Layer3)
  end for
end for
for i = 0 to number_trainset_entries2 do
  asso_train(Layer1_activation(i-1),Layer1_activation(i))
end for
for i = 0 to number_trainset_entries3 do
  asso_train(Layer2_activation(i-1),Layer2_activation(i))
  asso_train(Layer1(i),association Layer2(i))
end for
```

The black-board prediction needs more entropy on the output vector side to render the system able of predicting multiple next activation states with single retrieval operation. In the case of multi-step forecasting the dimension of the output vector is $n \cdot m \cdot k_\theta$ consisting k_θ ones: exactly analog to the input vector representation. Overlapping of events is not necessary in this association scheme, it is sufficient to learn the association of patterns starting at time step t to the pattern starting at time step $t + k_\theta$.

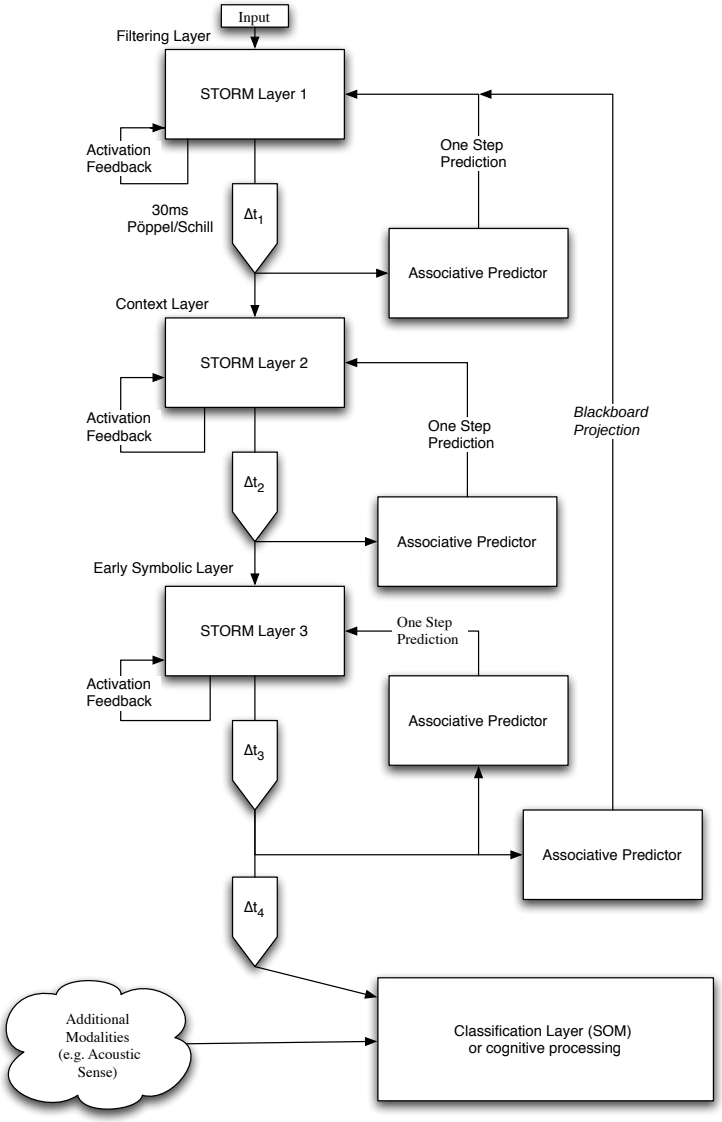

Figure 9.5: Final Spatio-Temporal STORM model with local and blackboard prediction loop.

Additionally to the motion perception and prediction part of our model, we added a kind of *cognitive* layer which is autonomous responsible for a classification of the patterns of the highest processing layer. This layer can be defined as e.g. a learning vector quantification network (LVQ) or again as a SOM layer. The small size and the small number of vectors in our training and testing data makes it impossible to conduct real classification task experiments. We can only evaluate the mere ability to classify correctly.

And in fact, classification worked. We will now discuss some issues on how to design the prediction mechanism and on the capacity of those network structures.

9.6 On the Prediction Capacity of stSTORM

By introducing a threshold that determines a significance level of a neuron and setting a neuron to a refractory state while being activated stronger than the threshold, it is possible to give a delimiting value of the maximum number of activation patterns. The size of a stSTORM layer is $n \times m$ neurons multiplied by the factor k_θ caused by the transcoding process. The number of activated neurons is implicit defined by the feed back factor α which results in a system response, in our case of $\alpha(1-\alpha)^k$, and the significance threshold θ_s. Given these elements we get a combinatorial problem: First determine the number of possible active neurons in one time frame: Determine the value of k_θ for which

$$k_\theta = min_k | \theta_s \geq \alpha(1-\alpha)^k \tag{9.5}$$

holds. So we have to draw k_θ activation values out of the pool consisting of $n \cdot m$ neurons multiplied with a transcoding factor representation based on k_θ. We are looking for all possible permutations we can draw these k_θ activation sequences in an ordered fashion out of the fixed pool simultaneously meeting the constraints of the transcoded activation vector:

$$(n \cdot m)^{\underline{k_\theta}} = \prod_{i=0}^{k_\theta - 1} ((n \cdot m) - i) \tag{9.6}$$

The term $(n \cdot m)^{\underline{k_\theta}}$ is a falling factorial, it defines the maximum number of patterns a single STORM layer is able to produce in a generative process.

Having the limit of possible patterns it is possible to calculate the maximum accurate prediction capability of the connected associative memory. Binary neural hetero associative memories were investigated quite deeply by G. Palm [PB88] [PSSS97] and F. Sommer [PS95] especially with the focus on pattern storage capacity and the definition of the term *sparseness* in relation with neural associative memories.

The limitations on the number of possible ones in the input, respectively in the output vector shifts the limit of maximal storable patterns a bit to the theoretical maximum as not all possible permutations of input vectors are used.

As a hetero-associative memory is tolerant to errors, a matrix entry – which in fact is an information bit– has not the information capacity 1 like in a standard storage structure, it has a capacity of $0.69 = \ln 2$.

The maximum accurate prediction capability of a stSTORM is bound by the number of patterns the associative memory is able to store. Using a

binary hetero associative memory makes it determinable, as the capacity C of neural associative memories has been analyzed by Günther Palm and Fritz Sommer [PS95]. To calculate the capacity we have to define some symbols: $p := pr[x_i = 1] \forall i$ for all possible input vectors and $q := pr[y_j = 1] \forall j$ respectively for the output vectors. δ is a quality factor. The smaller δ the lower the probability of a wrong 1 at any position of the output vector. Then the capacity C is defined by:

$$C = \ln 2 \frac{-\ln q}{-\ln q - \ln \delta} \tag{9.7}$$

Whereas q is defined as the probability of having a 1 in the output vector at position i. Then for $q \to 0$ the fraction above converges to $\ln 2$.

$$C \approx \ln 2 \tag{9.8}$$

A error tolerant memory is not able to reach the capacity of 1 bit per storage element. It only reaches the relative capacity of $\ln 2$. As long as the average number of ones in the memory does not reach $\ln 2$, retrieval is possible without any error.

Given the dimensions of the STORM layer and the tuple length of the transcoding, the associative memory matrix size is $(n \cdot m \cdot \ k_\theta)(n \cdot m)$.

The resulting maximum number of stored patterns M is:

$$M = \frac{\ln 2}{pq} \tag{9.9}$$

whereas p is the probability of a 1 in the input pattern and q the probability of a 1 in the output pattern. The probability of a 1 in q is simply:

$$q = \frac{1}{n \cdot m} \tag{9.10}$$

as only a single one is allowed to occur. The probability of a 1 in the input vector p is dependent on the number of possible representations, the falling factorial.

$$p = \frac{1}{n \cdot m^{\underline{k}}} \tag{9.11}$$

The maximum number of storable patterns is a theoretical limit of the associative memory we use for prediction. As often many similar patterns occur, the occupancy of the memory matrix is not equally distributed. The retrieval error increases in the areas with a higher density of ones. The performance of our prediction mechanism though is strongly bound to the input data.

Chapter 10

Experiments

In this chapter we introduce and discuss the experiments conducted with our model. Several model configurations were tested to ensure the correct behavior of the overall model. The first tests address only a single STORM layer with one prediction layer realized by a binary hetero-neural associative memory. This single-layer abstraction model was tested using time series, whereas the multi-layer model was tested on more motion perception like data gathered with a gesture recognition device.

10.1 Laser Time Series

The most secure way to test a system is to test its single functional elements. The smallest possible functional element of the introduced motion perception system consists of one STORM layer with an attached prediction network, realized through a binary hetero-neural associative memory (see sketch Figure 10.1). These two networks represent a single abstraction layer of our multi-layer network. We used a sinusoidal signal to tune the parameters for a one dimensional time series. Time is coded implicit by the fixed sampling rate the data sets were generated with.

In this experiment series we use a chaotic time series. The laser time series. It is widely used for benchmarking time series forecasting models (used e.g. in [WG94]) and used in this work to test the one step prediction capability and the multi step prediction capability of the one layer STORM assembly. Additionally we crosschecked the STORM layer based configuration with a SARDSTORM based system. As SARDSTORM is only able to represent single state transitions STORM should give a big advantage on data which has context distributed over many time steps, as the recurrent weight adaptation ensures the persistence of past activation events. One neuron in STORM is learning more of a concept of the past view time steps, whereas as stated SARDSTORM only represents state transitions.

Network Setup:

- $\alpha = 0.5$ which evokes a four time-step event horizon.
- Gaussian neighborhood
- 50×1 neurons first experiment
- 100×1 neurons second experiment
- 150×1 neurons third experiment
- Adaptive threshold handling in the associative memory

Comparing the test runs with each other shows that the 150 neuron experiment shown in Figure 10.5, yields a much better prediction ability for one step forecasting on the 1000 entry sized time series than the other runs. We only have half the number of neurons compared to "state transitions" in the data set which indicates that the network learned some concepts. The ultra-short-term memory lasts with fixed $\alpha = 0.5$ for four time steps. So each prediction relies on the combination of the four past activation events to predict the next possible active neuron.

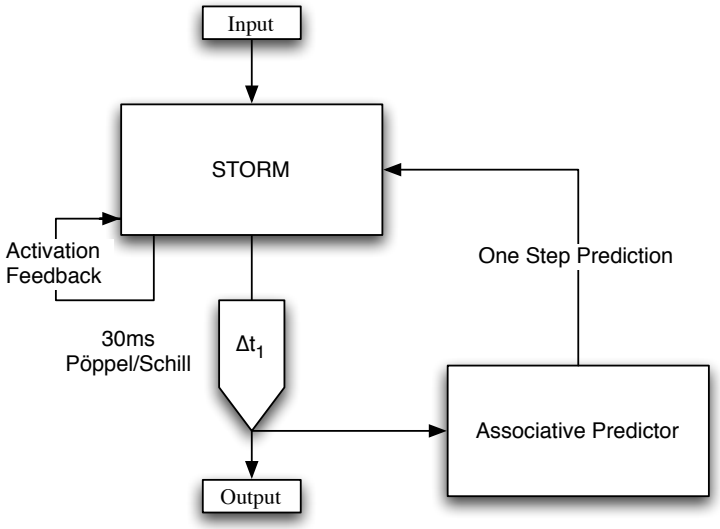

Figure 10.1: Single layer stSTORM.

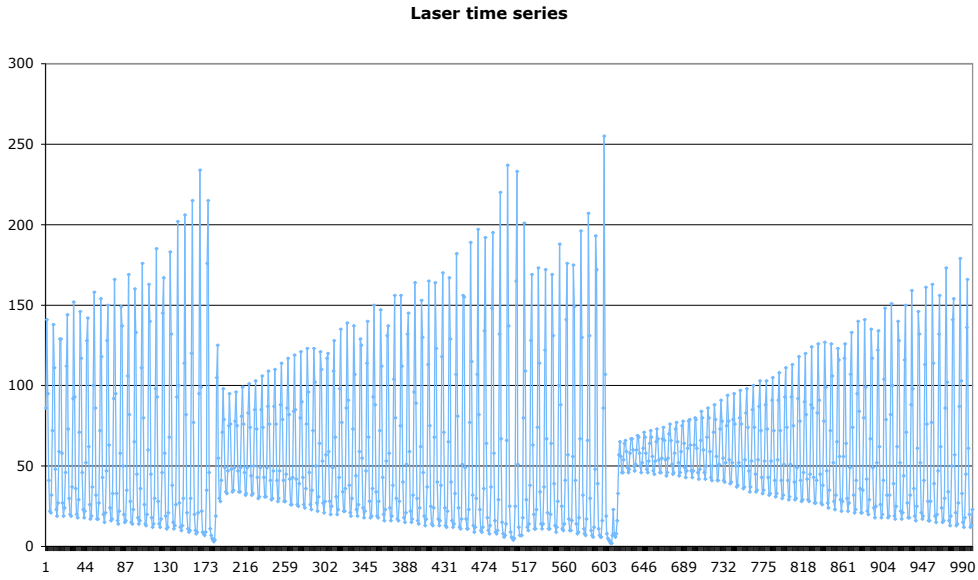

Figure 10.2: Laser time series

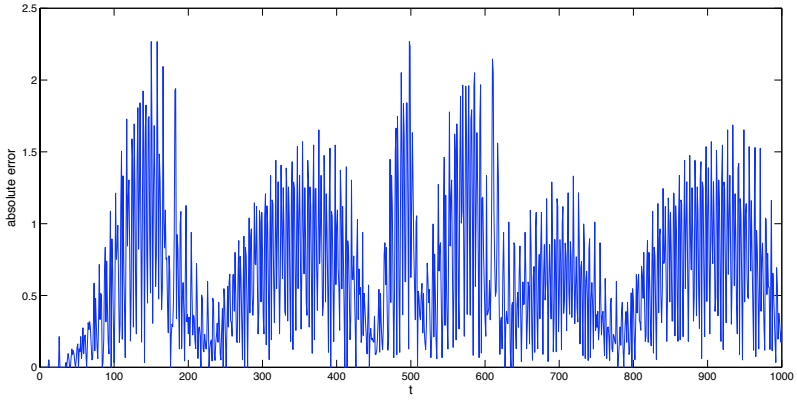

Figure 10.3: Single 50 neuron layer STORM with open prediction after 100 starting values $\alpha = 0.5$.

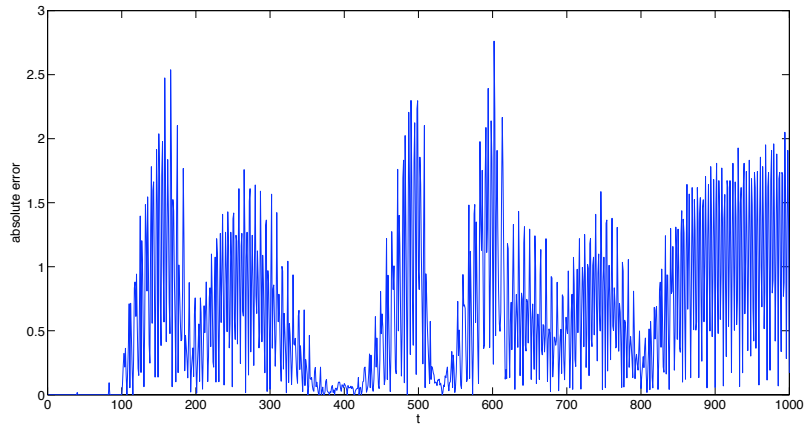

Figure 10.4: Single 100 neuron layer STORM with open prediction after 100 starting values $\alpha = 0.5$.

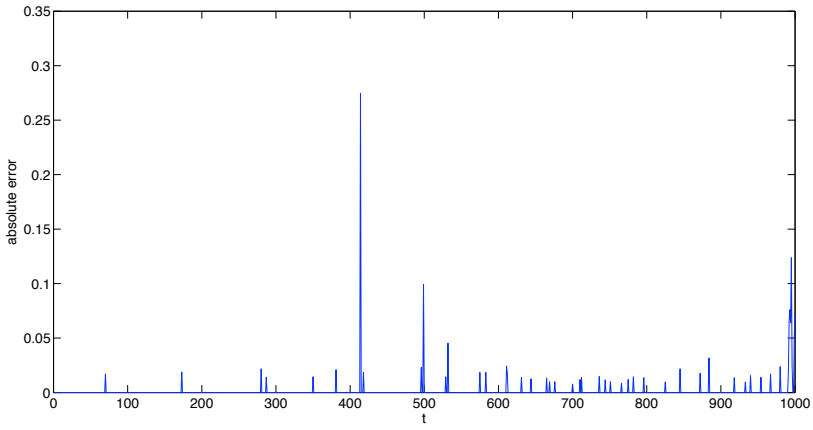

Figure 10.5: Single 150 neuron layer STORM with open prediction after 100 starting values $\alpha = 0.5$.

The number of neurons affects the networks processing ability in two ways. The higher the number of neurons, the better the approximation of the time series. Additionally, the higher the number of neurons, the bigger the memory matrix of the prediction network. These two parameters affect the assemblies performance crucially as one can see in the prediction error we plotted for a 50 neuron network assembly in Figure 10.3, a 100 neuron assembly in Figure 10.4, and a 150 neuron assembly in Figure 10.5. The high

prediction error in the smaller networks where mostly caused by the small number of neurons, the size of associative prediction memory has only been to small for the case of 50 neurons. The memory matrix of the associative prediction network was loaded not sparse enough. Testing the 100 neuron assembly the first 100 predicted values show only a small prediction error, followed by chaotic oscillating values.

We also run experiments with e.g. $\alpha = 0.7$ and higher, to illustrate the important influence of the activation feedback to the temporal processing capability. As easily can be seen, a temporal memory shaped with the feed-

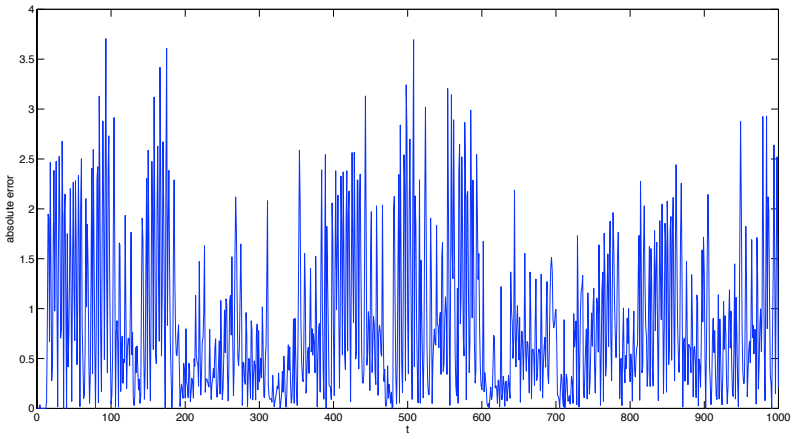

Figure 10.6: Single 150 neuron layer STORM with open prediction after 100 starting values $\alpha = 0.7$.

back factor $\alpha = 0.7$ simply is to short to learn sufficient context within the laser time series.

The 150 neuron sized assembly with $\alpha = 0.5$ shows an expected behavior in the last third of the predicted sequence caused by similarities between the second part and the third part of the sequence. Differences between the second and the third part of the sequence occur at the position around time step 500. The prediction mechanism is not able to distinguish between the possible development in the second part and the third sweep of the series. Even the context preserving weight adaption is not helping in this case. Simply to many time steps are involved and the context horizon in this case was 3 time steps. To point the necessity of context preserving learning out we introduced a non recurrent weight adaption scheme to our STORM network and named it after the inspiring SARDNET SARDSTORM.

SARDSTORM showed a very poor, even not existent open prediction capability as showed in Figure 10.7. Even in one step prediction SARD-STORM is not able to predict correctly, if context is necessary to determine

the right next value. Once again the similarity of the second and the third part of the time series causes problems. As no context information was learned, the neural assembly has no chance to predict the correct values in the third sweep, in contrast to our STORM assembly whose performance is shown in Figure 10.8.

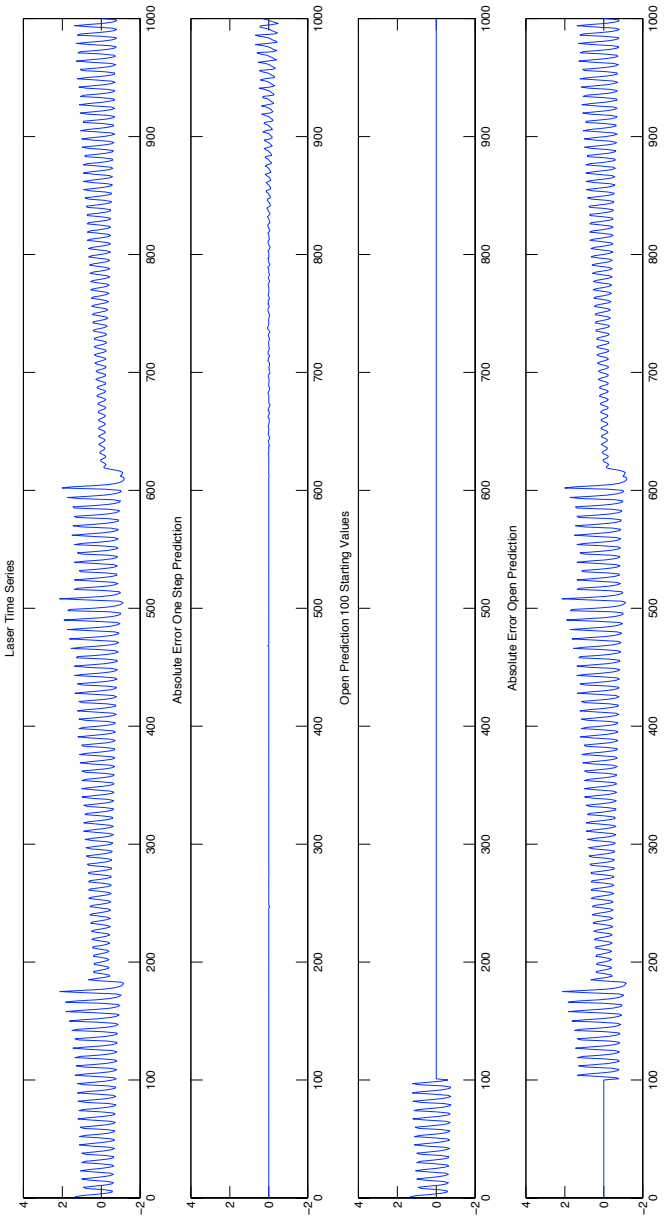

Figure 10.7: Single 150 neuron layer SARDSTORM with one step prediction error, open prediction after 100 starting values and absolute prediction error of the open prediction.

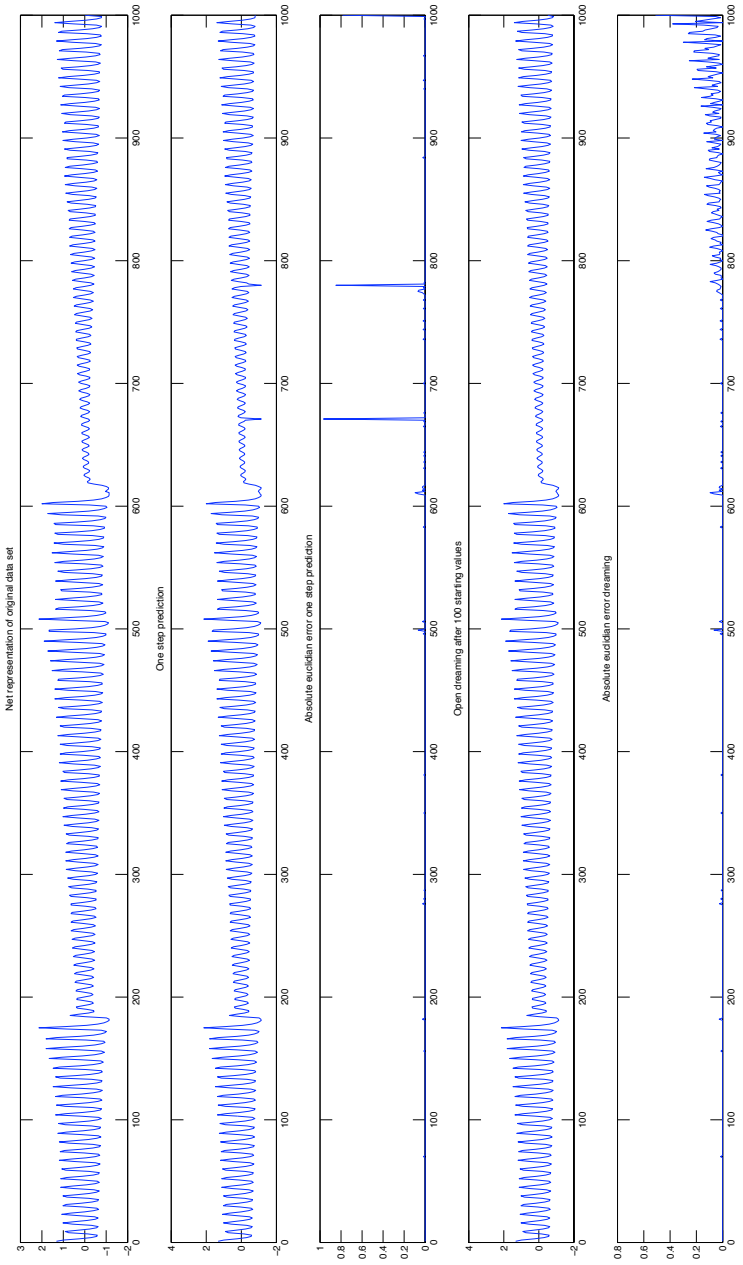

Figure 10.8: Single 150 neuron layer STORM with one step prediction, one step prediction error, open prediction after 100 starting values and absolute prediction error of the open prediction.

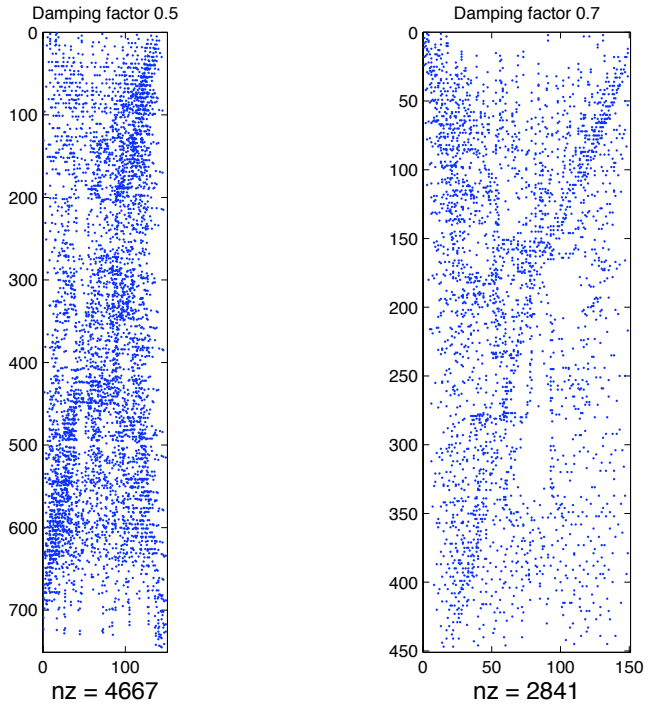

Figure 10.9: Laser time series with one step trend prediction. Map size 150×1: Prediction memory population for $\alpha = 0.7$ and $\alpha = 0.6$. nz: non zero elements.

10.2 Qualitative Motion Vector based Data

The DFG priority program Spatial Cognition had the aim to provide amongst other things symbolic spatio-temporal descriptions of motion paths. One thesis in this research program dealt with a qualitative description of motion [Mus00]. The formalism allowed a generalization of motion but was not able to predict future motion elements. Our multi-layer motion processing model is capable of prediction; so the step to link qualitative motion vectors (QMVs) with multi-layered STORM is quite obvious. In the next lines you will find a simplified *quasi* qualitative motion description. We will use this input data to illustrate some behaviors of the model to motion. The input vector consist basically of direction changes measured in degrees and the velocity, which is subdivided in numerical representations of slow, medium and fast. The distance is implicitly encoded due to the fact that we assume a discretized input sampled at a frequency of 30Hz.

Due to the small number of test stimuli at least the first processing layer

Direction	Speed	Label
90	1	right_slow
90	1	right_slow
90	2	right_medium
90	1	right_slow
180	1	down_slow
180	2	down_medium
180	1	down_slow
270	1	left_slow
270	2	left_medium
270	1	left_slow
0	1	up_slow

Table 10.1: Numerical representation of QMV vectors

has to be trained with visual stimuli consisting of arbitrary combined motion atoms to form a orientation selective mapping. Nonetheless the QMV based data has to view real features to be learned well.

Provided with this first filtering layers we are able to observe the behavior of the higher processing layers. The second processing layer joins dependent on the predetermined feedback factor, motion atoms to motion compounds. The third layer combines these compounds to more complex motion path elements.

The test set consisted of a sequence of QMV trajectories sketched in Figure 10.10: QMV1, QMV2, QMV1, QMV3, QMV2, QMV4, QMV3, QMV4.

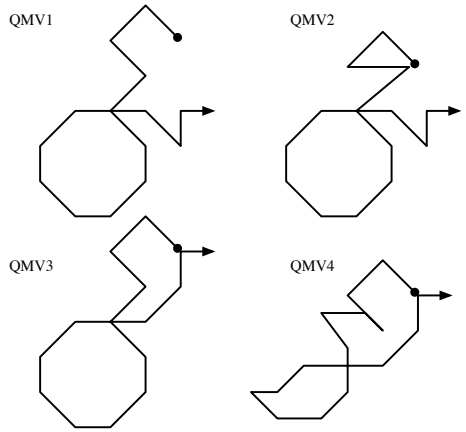

Figure 10.10: Simple QMV trajectory examples, also used in psychophysical experiments

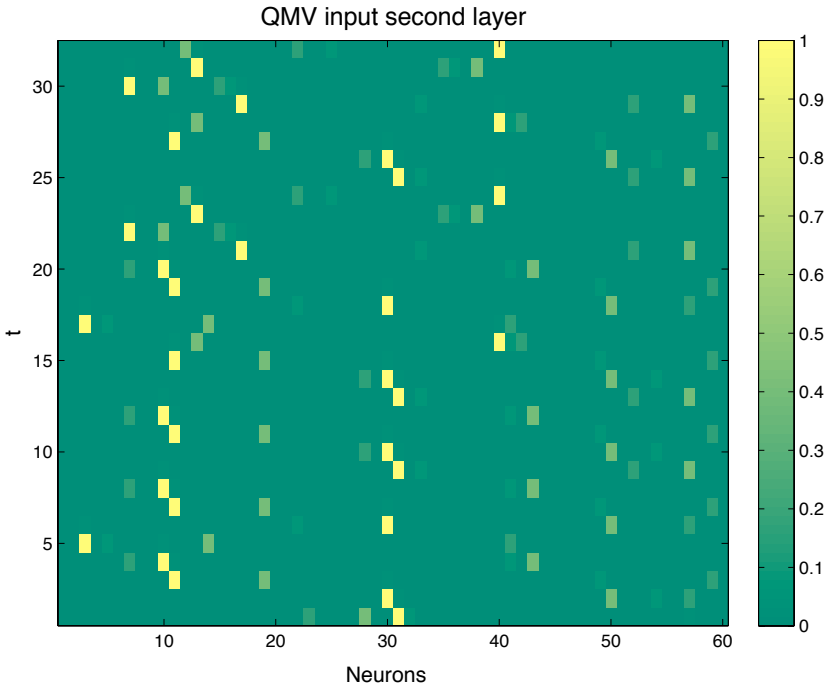

Figure 10.11: Data propagated from the first to the second processing layer. Each row is an activation snapshot joining the last four time steps. Color indicates the activation strength of the referred neuron.

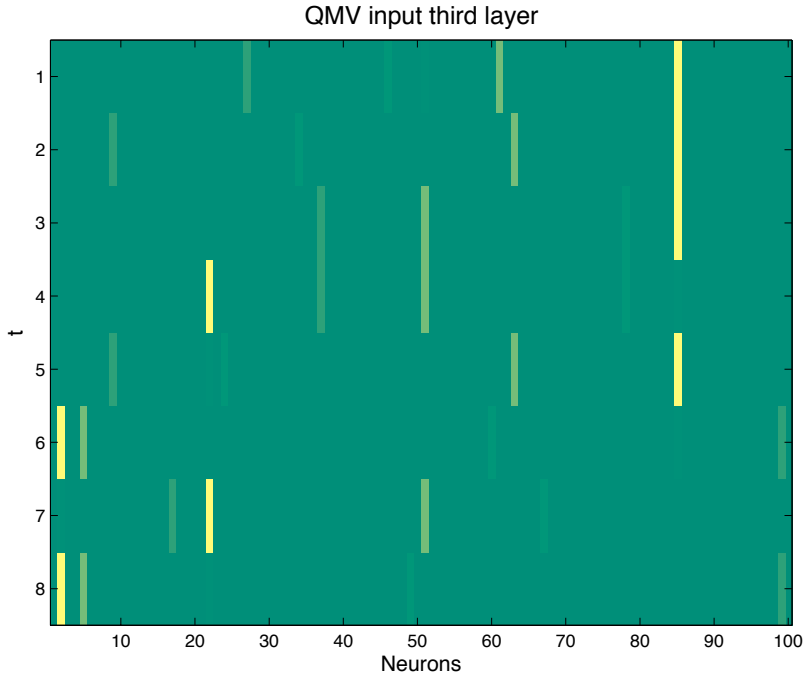

Figure 10.12: Data propagated from the second to the third processing layer. Each row is an activation snapshot joining the last four time steps. Color indicates the activation strength of the referred neuron.

Generally, one would expect that a cyclic motion performed with a somehow correct frequency will induce some attractors on higher processing layers, and in fact attractors where found on the second processing layer in our three layer model. The first layer acts like an compound of orientation selective filter, whereas each neuron reacts on a stimulus of a special direction and velocity. We derive a more abstract description of motion by the summation of several activation events before propagating an activation pattern to the next higher processing layer. The next higher level acts exact in the same way. As the hierarchical level gets higher, the represented motion compounds get more abstract and more complex.

We were able to observe exactly the same effects like in the psychophysical experiment on the same stimuli. Small changes in the trajectory not effecting a geometrical structure like the part of the motion path which is similar to a circle are not detected by the system. QMV1 versus QMV3 versus QMV2 are not discriminable. QMV4 on the other hand is discriminable quite easily, as one geometrical structure, the circular motion part is affected over a temporally longer period. The changes in the first three motion paths are also temporally to short, to have an big impact to the system on a higher

–in this case already on the second– processing layer. It would of course be possible to outperform humans recognition abilities by prolonging the temporal memory of one neuron. In that case one neuron would be able to learn the concept of a complete trajectory within one activation period and the second layer would represent one whole trajectory by the firing of one single neuron and therefor the chance of not learning a representation but memorizing one gets really high.

To enter a more realistic scenario we derived some motion data in collaboration with Matthias Kranz PhD student at the embedded interaction group, chair for media informatics at the Ludwigs-Maximilian-University Munich.

10.3 Gesture Data Set

These data sets were gathered with a hollow wooden cube, containing three orthogonal acceleration sensors (Figure 10.13). The sensor readings are transfered via wireless interfaces in real time to a PC unless the embedded processors do not have the necessary computing power to realize a neural network of the complexity of the introduced multi-layered STORM model. Some examples of simple gestures are plotted in Figure 10.14.

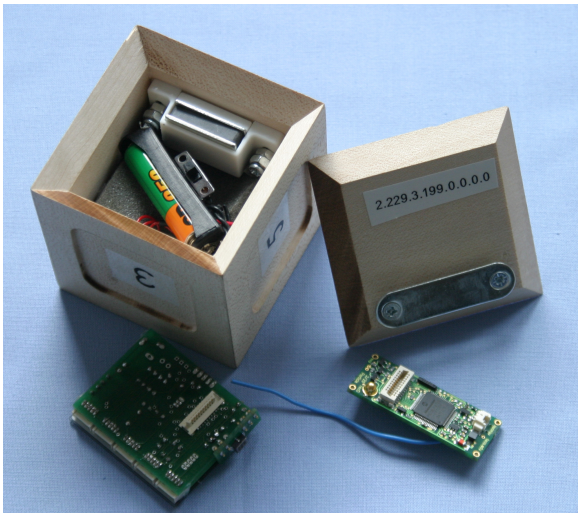

Figure 10.13: Gesture Cube

We set up a three layer model like sketched in Figure 9.5 to test the behavior of the model on the introduced gesture data. For training purpose we combined all gestures in a single data file plotted in Figure 10.17.

As one would expect, temporally stationary elements in the data file are reflected by an attractor like firing pattern of neurons. Whereas in this case stationary elements are characterized by a cyclic temporal pattern. Sinusoidal like input signals with fixed amplitude and frequency as a temporal constant signal are a simple example for a stationary signal. The STORM layer filters the signal and joins several time steps to one activation pattern.

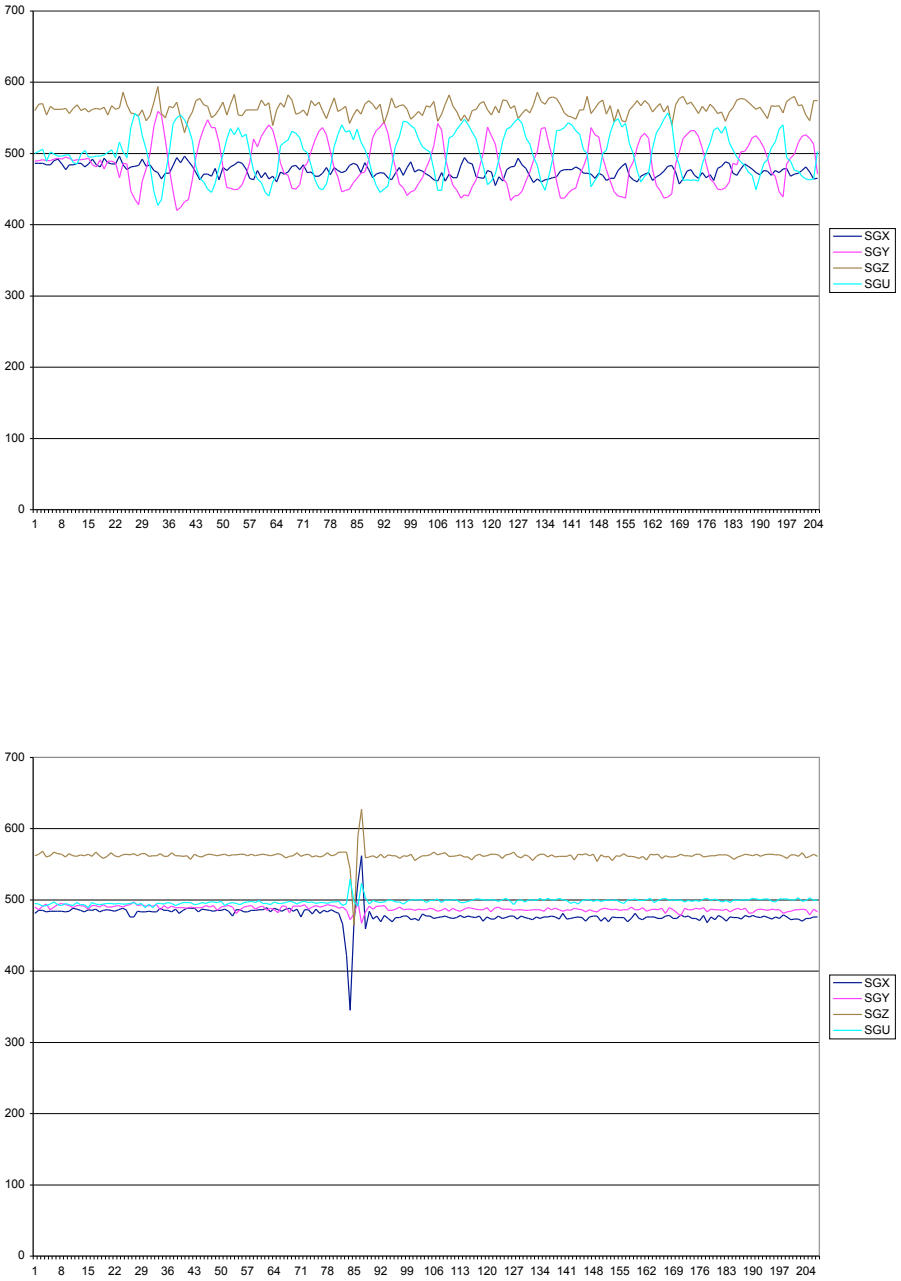

Figure 10.14: Sensor reading of left/right shaking and shift backward motion.

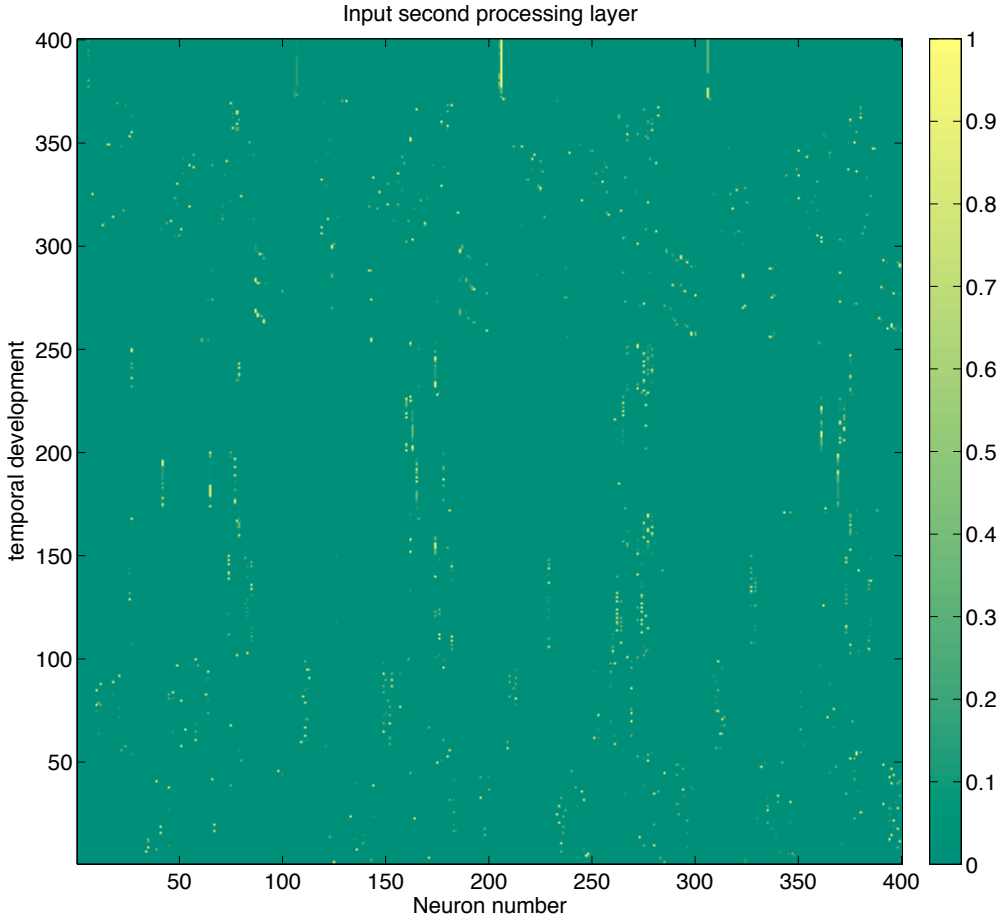

Figure 10.15: Data propagated from the first to the second processing layer. Each row is an activation snapshot joining the last four time steps. Color indicates the activation strength of the referred neuron.

Generally, all alternate processing layers again combine several time steps of activation events of the earlier stages to a new temporally more coarse activation pattern. The input representation of the third layer is shown in Figure 10.16. It is more difficult to find the stationary signal parts here, despite for the last time steps. The input data set shows at step 1500 to the end of the sequence in Figure 10.17 four singular signal lines. These lines are represented as one activation pattern in Figure 10.16 at about time step 92 to 100.

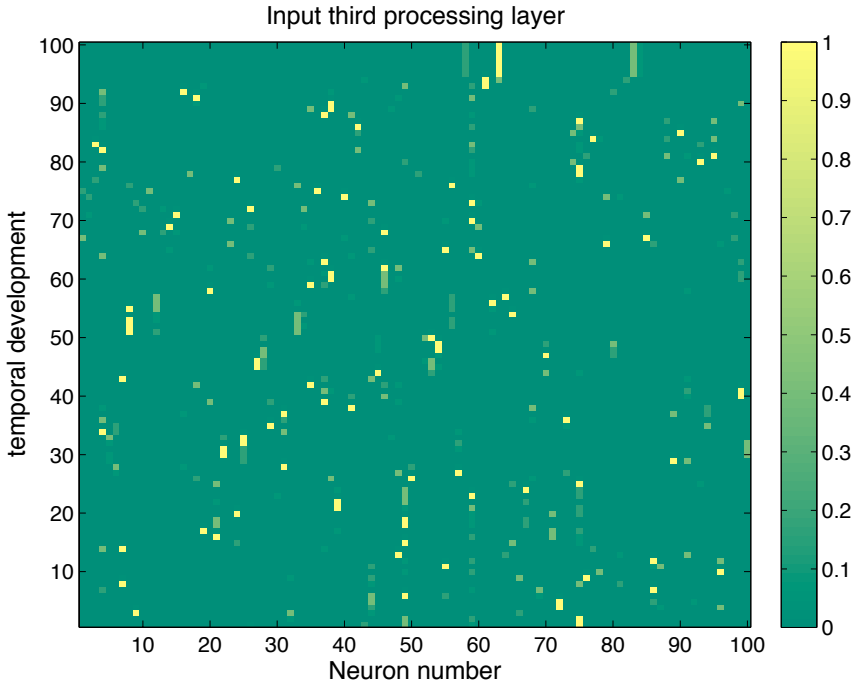

Figure 10.16: Data propagated from the second to the third processing layer. Each row represents an activation snapshot of the last four time steps. Color again indicates the activation strength.

The experiments showed that the spatio-temporal STORM model is able to abstract data from a provided input stream automatically. This behavior is similar to a filtering of the signal, whereas our model additionally stores context information which is important for prediction issues. As the experiments on the laser time series showed the system is robust on signal drop outs through its prediction mechanism.

It is possible to place a pattern classification algorithm on top of our model, to classify the gestures. Technically every algorithm would be suitable. In the motion processing system of the real brain, the next processing step would involve cognition. A conscious or unconscious tagging of the perceived motion stimuli to form and generate trajectories in mind.
We will continue on our discussion in the conclusion chapter.

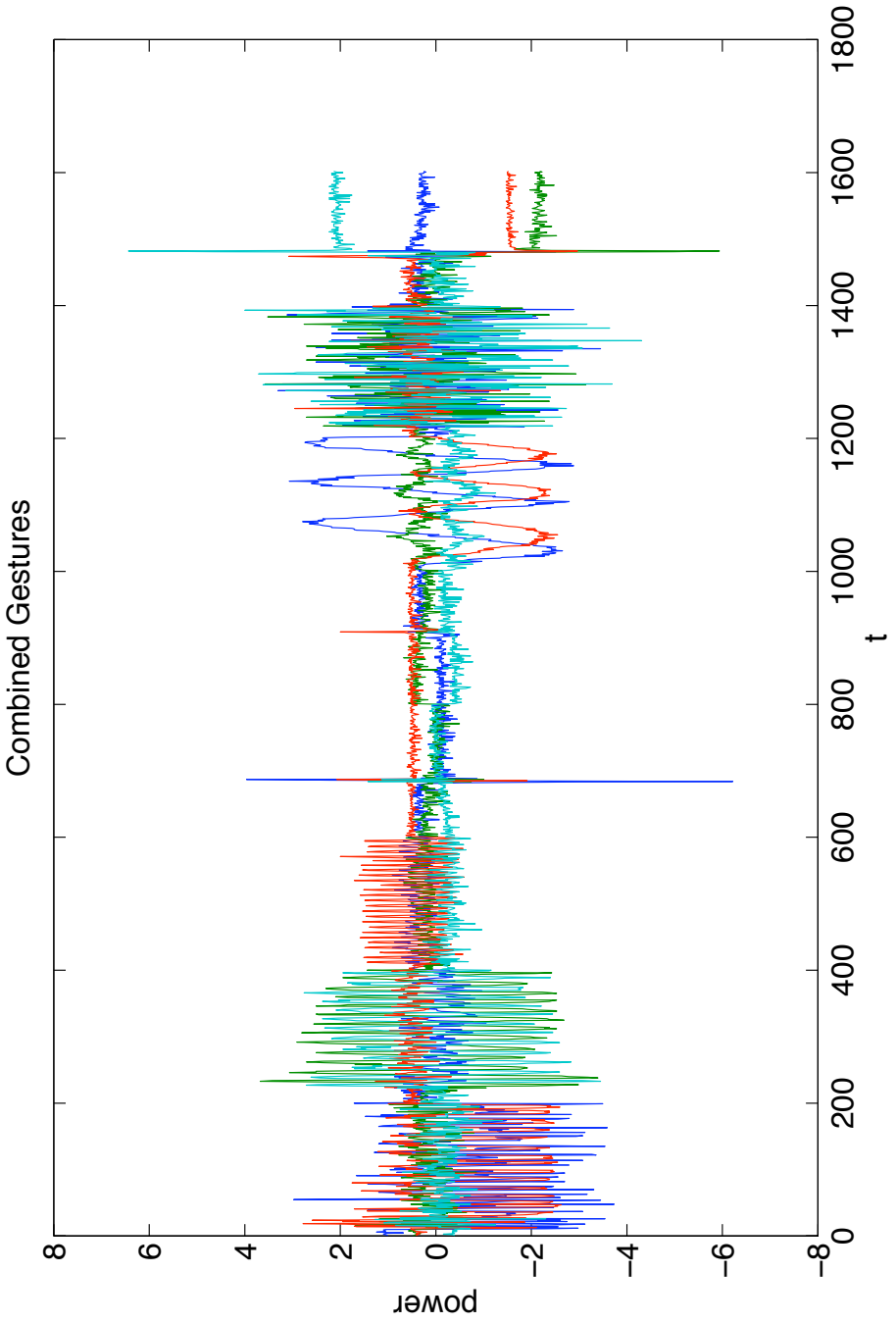

Figure 10.17: Test input data set derived from a combination of all gesture data sets.

Chapter 11

Conclusion and Discussion

This thesis introduced a novel motion processing and prediction model. The realization of the model is the result of the combination of findings from different research schools like psychophysics, neuro-anatomy, connectionism and within connectionism, computer science. Defining the model was not possible by using only established models. We had to define novel flavors of recurrent self-organizing maps for the single processing layers. Also the interaction between the processing layers was defined in a suitable way to meet the demands of an orthogonal access of space and time along the whole system and in each of the processing layers. We will now recapitulate on each part of our system to emboss the novelties and to discuss the results of this thesis.

- **RSOM:** We developed a training method which ensures temporal ordering and prevents poor topological development of the map. The training method is based on selecting subsets of the sequence. The length of the subsequences is delimited by the *event horizon* of the first firing neuron of the subsequence. The first element of a subsequence is selected randomly. This pseudo random scheme is done the first ten percent of the training phase. The rest of the training is done in ordered fashion, as the network is able to unfold itself within the first ten percent training cycles.

  The second novelty is the processing of the activation of the map. We are using the activation configuration of the complete map for further processing, not only the actual winning neuron.

- **Strict temporally ordered map, STORM:** In contrast to the SARDNET variant of the self-organizing map we use a recurrent weight adaption within the learning algorithm. Additionally we defined the event horizon to determine the time span a neuron is refractory and taken out of the pool of competing neurons. SARDNET removes each neuron winning the competition for the best match. The most import-

ant advantage of our definition is that the context information is not only dependent on the last occurring state transition.

Similar to SARDNET is the use of a standardized activation of a competition winning neuron. The use of this activation is different though. We use the activation of the neurons to generate an activation pattern. The activations of all neurons are combined to a single vector, which is intended to be fed to the next higher processing layer. Additionally the activation vector is stored –in a transcoded representation– in a neural associative memory.

As we use a recurrent weight adaptation, STORM stores the temporal development of several time steps. This recurrent weight calculation results in combination with the associative prediction memory in a much better prediction capability of a STORM associative memory assembly.

- **Stochastic Spiking STORM** Not yet realized in a multi-layer context, we showed a possibility to go further towards a biological more plausible self-organizing map definition. As the activation is calculated in temporal distributed fashion a single stochastic spiking STORM works on a temporally finer resolution as STORM. Adding up the time steps of one activation event of a neuron results in a STORM neuron. The stochastic spiking definition might be useful for biologically plausible representation of temporal processes.

- **Spatio-Temporal RSOM:** The piloting multi-layer model, we realized. Inspired by LAPS we introduced the orthogonal access memory within each processing layer. Also the information propagation from one processing layer to the hierarchically next higher was changed as the use of recurrent neurons made it possible to represent several time steps in an activation pattern derived by the enumeration of all activation values of all neurons of one layer. Not quite pleased with the feed-forward network for the prediction of future input values and the limited temporal resolution of the prediction network we developed a more sophisticated multi-layer model.

- **Spatio-Temporal STORM:** Analyzing spatio-temporal RSOM we recognized that the temporal ordering in the activation pattern can be broken. This resulted in the above discussed STORM network and with the replacement of the RSOM layer in the spatio-temporal RSOM assembly.

  Additionally we searched for a more sophisticated prediction mechanism. With the introduction of neural hetero-associative memories we added a efficient robust computing structure. These memories act like an indexing structure similar to retroinjecting lines in real brains,

lowering the threshold in those neurons which are most likely to be activated in the next time step.

It should be stressed that already the one layered stSTORM network has the computational power of a LAPS network consisting of two SOM layers connected with a feed-forward network for prediction purpose. Our model is more efficient in time and space.

- **SARDSTORM:** To point the advantage and the necessity of a recurrent weight adaptation scheme out, we defined a network similar to SARDNET. SARDSTORM uses two elements of SARDNET. First the standardized activation value of winner neurons and second, a non recurrent weight adaptation identical to standard SOM. In contrast to SARDNET, SARDSTORM puts a once activated neuron back into the pool of competing neurons after an absolute refractory period like in our STORM network definition.

So far we only reflected the technical algorithmic side of our work. The algorithmic realization of the model is more or less only the tool for a motion processing and prediction model. All the novel definitions in our model were guided and inspired in the results of psychophysical experiments done in collaboration within the research group of Kerstin Schill. We finally introduced a model which fulfilled all demands we derived in interdisciplinary discussions.

The model shown in Figure 9.5 already shows the next steps we intend to add to our model. We already experimented with hand tagged classification, yet a cognitive processing stage needs more attention. So far the model provides the demand of an orthogonal access memory, various feedback and retroinjecting lines between several processing stages, and therefor the ability to process and predict motion in a hierarchical manner. Nonetheless our model is a highly sophisticated preprocessing mechanism providing proper input to higher cognitive processing, this already is one of the topics of the planned future work.

Chapter 12

Future Work

In the attempt to realize a visual motion processing model based on psychophysical experimental findings we realized an artificial neural assembly. This model was designed with biology in mind but biology was only the second most important source of inspiration. To tune the model to more biology like representations we suggest to incorporate the prediction indexing structure realized by the neural associative memory so far, into the STORM layer. This additional structure could be realized be a feed forward network sitting beneath each of the STORM layers neuron. Learning might be done with error back-propagation. Having these two different styles of training in one assembly brings on idea into account, which addresses two stages of learning. We argued already for this twofold learning process earlier. The adaptation of the STORM layer is realized by a competition process in which the weights are adapted using the parameters learning rate and neighborhood. This part of learning in an dynamic processing system is in our believe more analog to the evolutionary ordering process of neural cells to functional areas in the maturation of a fetus. Learning in the post-fetal stage –at the age of two months a new born baby has the maximum number of interconnections between neurons– is in our belief more like hebbian learning. This second learning stage, subdivided from the evolutionary learning process, is reflected in the training of the dynamic extension, the feed-forward net of each STORM neuron.

Another possible extension addresses the stochastically spiking STORM. In the search for a more biologically realistic model still adaptable in reasonable time we argue for a replacement of the exponential trace memory by a gamma trace memory. The gamma memory establishes a more realistic onset behavior of a single neuron.

Beside the possible model improvements we can also give some ideas on how our proposed motion processing model might be used in technical systems like:

- Driver assistant systems

- Reactive robot motion planning
- Dynamic pattern analysis and concept generation
- Saccadic eye movement analysis classification concept generation
- Cognitive motion planning
- Realization of inter-modal connections

One rather interesting point would be a distributed realization of our model. As the calculation in self-organizing map based model is highly local it is possible to construct a parallel version quite easily. A diploma thesis related to this work realized a parallel version of standard self-organizing maps based on the PVM toolkit (parallel virtual machine). Based on this diploma thesis it would be possible to develop parallel versions of all self-organizing maps based models we introduced.

It would be even simpler to distribute only complete layers over a parallel computing facility, as all layers process their information independently and the interaction happens not every calculation time step.

As the implementation is based on matlab, we cannot say anything about real time capability. Though retrieval, compared to training is blazingly fast as a complete data set of 2000 vectors is retrieved within three seconds which should be enough taken the flicker fusion frequency into account. But possibly an implementation in C or C++ could fulfill realtime capabilities better. So one next steps on the implementation side should address a redesign of the software in a more efficient language to test, if the system can be used in real world applications like the discussed Driver assistant systems, or in reactive robot motion planning.

The most important step in continuing the research is the development of a cognitive processing stage which is able to process the motion data generated by our stSTORM model to define motion plans autonomously by observing its own percepts.

There are still many more possibilities in extending and improving the here proposed stSTORM. Hopefully this work inspired some further development in visual motion processing and prediction.

Bibliography

[AB85] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A*, A2:284–299, 1985.

[And97] R. A. Andersen. Multimodal integration for the representation of space in the posterior parietal cortex. *Phil. Trans. R. Soc. Lond. B*, 352:1421–1428, 1997.

[Bai05] V. Baier. Motion perception with recurrent self-organizing maps based models. In *IJCNN 2005 Conference Proceedings*. IEEE, 2005.

[Bay87] D. A. Baylor. Photoreceptor signals and vision. *Invest. Ophthalmol. Vis. Sci.*, 28:34–49, 1987.

[BB03] V. Baier and W. Brauer. Sequence processing with recurrent self-organizing maps based models. In M. Reischl R. Mikut, editor, *Proceedings 13. Workshop Fuzzy Systeme, FZKA-6900*. Forschungszentrum Karlsruhe, 2003.

[BKRB00] V. Baier, K.Schill, F. Röhrbein, and W. Brauer. Processing of spatio-temporal structures: A hierarchical neural network model. In C. Freksa, editor, *Dynamische Perzeption*, pages 223–226. Workshop der GI-Fachgruppe Bildverstehen, G. Baratoff, H. Neumann, Berlin 2000.

[BQ01] O. Braddick and N. Qian. *The Organization of Global Motion and Transparency*, chapter Computational, Neural, and Ecological Constraints, pages 86–112. Springer-Verlag, 2001.

[Bri97] G. Briscoe. *Adaptive Behavioural Cognition*. PhD thesis, Curtin University of Technology, School of Computing, September 1997.

[Bul01] J. Bullier. Integrated model of visual processing. *Brain Research Reviews*, 36:96–107, 2001.

[CG87] G. Carpenter and S. Grossberg. A massively parallel architecture fo a self-organizing neural pattern recoginition machine. *Computer Vision, Graphics and Image Processing*, 37:54–115, 1987.

[DD86] H. L. Dreyfus and S. E. Dreyfus. *Mind over Machine*. Free Press, 1986.

[dVP92] V. de Vries and J. Principe. The gamma model – a new neural network for temporal processing. *Neural Networks*, 5(4):565–576, 1992.

[ESR+00] A. Eisenkolb, K. Schill, F. Röhrbein, V. Baier, A. Musto, and W. Brauer. Visual processing and representation of spatiotemporal patterns. In C. Freksa, C. Habel, and K. Wender, editors, *Spatial Cognition II - Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications*, number 1849 in Lecture Notes in Artificial Intelligence, pages 145–156. Springer, Berlin, 2000.

[EZM+98] A. Eisenkolb, C. Zetzsche, A. Musto, W. Brauer, and K. Schill. Analysis and modeling of the visual processing of spatiotemporal information: Influence of temporal and spatial separation on the discrimination of trajectories. *Perception*, 27(supplement):188, 1998.

[Gar85] H. Gardner. *The Mind's New Science: A History of the Cognitive Revolution*. Basic Books, 1985.

[Glü90] H. Glünder. ΣΠ-networks for motion and invariant form analysis. In R. Eckmiller, G. Hartmann, and G. Hauske, editors, *Proceedings of the international conference on parallel processing in natural systems and computers*. Elsevier, Amsterdam, 1990.

[Gro86] S. Grossberg. Adaptive pattern classification and universal recording i: Parallel development and coding of neural features. *Biological Cybernetics*, 23:11–61, 1986.

[Hay94] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Macmillan, New York, 1994.

[Hay02] S. Haykin. *Adaptive Filter Theory*. Prentice Hall, fourth edition, 2002.

[HLD+98] N. Hadjikhani, A. K. Liu, A. M. Dale, P. Cavanagh, and R. B. Tootell. Retinotopy and color sensitivity in human visual cortical area v8. *Nature Neuroscience*, 3:235–241, 1998.

[HW77] D.H. Hubel and T.N. Wiesel. Functional architecture of macaque monkey striate cortex. *Proc. Royal Soc. Lond.*, 198:1–59, 1977.

[JM95] D. L. James and R. Miikkulainen. Sardnet: A self-organizing feature map for sequences. *Advances in Neural Processing Systems*, 7, 1995.

[Joh75] G. Johansson. Visual motion perception. *Scientific American*, 232:76–88, 1975.

[Koh97] T. Kohonen. *Self-Organizing Maps*. Springer, second edition, 1997.

[KVHK98] T. Koskela, M. Varsta, J. Heikkonen, and K. Kaski. Temporal sequence processing using recurrent som. In *KES'98 2nd Int. Conf on Knowledge-Based Intelligent Engineering Systems, Adelaide, Australia*, volume 1, pages 290–297, April 1998.

[Las51] K. S. Lashley. *The problem of serial order in behavior*, chapter Cerebral Mechanisms in Behavior, pages 112–146. Wiley, 1951.

[Len80] P. Lennie. Parallel visual pathways. *Vision Res.*, 20:561–594, 1980.

[MAGN85] J. A. Movshon, E. H. Adelson, M. Gizzi, and W. T. Newsome. The analysis of moving visual patterns. In C. Chagas, R. Gattas, and C. G. Gross, editors, *Study group on pattern recognition mechanisms*, pages 117–151. Pontifica Academia Scientiarum, Vatican City, 1985.

[MD85] J. Moran and R. Desimone. Selective attention gates visual processing in the extrastriate cortex. *Science*, 229:782–784, 1985.

[MP69] M. Minsky and S. Papert. *Perceptrons*. MIT Press, Cambridge MA, 1969.

[MSE+00] A. Musto, K. Stein, A. Eisenkolb, T. Röfer, W. Brauer, and K. Schill. From motion observation to qualitative motion representation. In Ch. Freksa, W. Brauer, Ch. Habel, and K. F. Wender, editors, *Spatial Cognition II*, number 1849 in LNAI, pages 115–126. Springer-Verlag Berlin, 2000.

[Mus00] A. Musto. *Qualitztive Repräsentation von Bewegungsverläufen*. PhD thesis, TU München, 2000.

[Nih73] Y. Nihei. A preliminary study on the geometrical illusion of motion path: the kinetic illusion. *Tohoku Psychologica Folia*, 32:108–114, 1973.

[NM91] A. Nomura and H. Miike. Field theory approach for determing optical flow. *Pattern Recognition Letters*, 12:183–190, 1991.

[NS81] A. Newell and H. Simon. *Communications of the Association for Computing Machinery*, volume 19, chapter Computer science as empirical inquiry: Symbols and search, pages 113–126. 1981.

[Pan74] A. Pantle. Motion aftereffect magnitude as a measure of the spatio-temporal response properties of direction sensitive analysers. *Vision Res.*, 14:1229–1236, 1974.

[PB88] G. Palm and T. Bonhoeffer. Patent for sparse coding in associative memories. *U.S. Patent No. 4 777 633*, Oct. 11 1988.

[PdVdO93] J. C. Principe, B. de Vries, and P. G. de Oliveira. The gamma filter- a new class of adaptive iir filters with restricted feedback. *IEEE Transactions on Signal Processing*, 41(2):649–656, 1993.

[PR73] T. Poggio and W. Reichardt. Considerations on models of movement detection. *Kybernetik*, 13:223–227, 1973.

[PRLN92] J. G. Proakis, C. M. Rader, F. Ling, and C. L. Nikias. *Adavanced Digital Signal Processing*. Macmillan, New York, 1992.

[PS95] G. Palm and F. T. Sommer. Associative data storage and retrieval in neural networks. In E. Domany, J.L. van Hemmen, and K. Schulten, editors, *Models of Neural Networks*, volume III, pages 79–118. Springer, New York, 1995.

[PS96] C. M. Privitera and L. Shastri. Temporal compositional processing by dsom hierarchical model. Technical report, International Computer Science Institute Berkeley, 1996.

[PSSS97] G. Palm, F. Schwenker, F. T. Sommer, and A. Strey. Neural associative memory. In A. Krikelis and C. Weems, editors, *Associative Processing and Processors*, pages 307–326. IEEE CS Press, Los Alamitos, CA, 1997.

[QdH05] F.-T. Qiu and R.von der Heydt. Figure and ground in the visual cortex: V2 combines stereoscopic cues with gestalt rules. *Neuron*, 47:155–166, 2005.

[RD02] E. T. Rolls and G. Deco. *Computational Neuroscience of Vision*. Oxford, 2002.

[RK86] H. Ritter and T. Kohonen. Self-organizing semantic maps. *Biol Cybernetics*, 61:241–254, 1986.

[RM86] D.E. Rumelhart and J.J. McClelland, editors. *Parallel Distributed Processing.* MIT Press, Cambridge, MA, 1986.

[Roj96] R. Rojas. *Neural networks.* Springer, Berlin, New York, 1996.

[RSB+03] F. Röhrbein, K. Schill, V. Baier, K. Stein, C. Zetzsche, and W. Brauer. Motion shapes: Empirical studies and neural modeling. In Ch. Freksa, W. Brauer, Ch. Habel, and K. F. Wender, editors, *Spatial Cognition III: Routes and Navigation, Human Memory and Learning, Spatial Representation and Spatial Learning*, chapter Spatial Representation, pages 305–320. Springer, 2003.

[Sch91] J. C. Scholtes. Recurrent kohonen self-organization in natural language processing. In T. Kohonen, K. Mäkisara, O. Simula, and J. Kangas, editors, *Artificial Neural Networks*, pages 1751–1754. Elsevier Science Publishers B. V. (North-Holland), 1991.

[Ski57] B. F. Skinner. *Verbal Behaviour.* Skinner Foundation, 1957.

[SZ95] K. Schill and C. Zetzsche. A model of spatio-temporal memory: the icon revisited. *Psychological Research*, 57:88–102, 1995.

[UM82] L.G. Ungerleider and M. Mishkin. Two cortical visual systems. In D.J. Ingle, M.A. Goodale, and R.J.W. Mansfield, editors, *Analysis of Visual Behavior*, pages 549–586. MIT Press, Cambridge, MA, 1982.

[Wat87] A. B. Watson. Efficiency of a model human image code. *J Opt Soc Am A*, A4:2401–2417, 1987.

[WG94] A. S. Weigend and N. A. Gershenfeld, editors. *Time Series Prediction: Forecasting the Future and Understanding the Past.* Perseus Books, 1994.

[WR92] H. R. Wilson and W. A. Richards. Curvature and separation discrimination at texture boundaries. *J. Opt. Soc. Am. A*, 9(10):1653–1662, 1992.

[ZB91] C. Zetzsche and E. Barth. Direct detection of flow discontinuities by 3D-curvature operators. *Pattern Recognition Letters*, 12:771–779, 1991.

[Zek74] S. M. Zeki. Functional organization of a visual area in the posterior bank of the superior temporal sulcus of the rhesus monkey. *J. Physiology*, 236:549–573, 1974.

[Zek78] S. M. Zeki. Functional spezialisation in the visual cortex of the rhesus monkey. *Nature*, 274:423, 1978.