Ulrich Ramacher
Christoph von der Malsburg
Editors

On the Construction of Artificial Brains

Springer

On the Construction of Artificial Brains

Ulrich Ramacher,
Christoph von der Malsburg (Eds.)

On the Construction of Artificial Brains

Springer

Editors

Prof. Dr. Ulrich Ramacher
Head of System Engineering
COM SDR
Infineon Technologies AG
81739 Munich
Germany
Email: Ulrich.Ramacher@infineon.com

Prof. Dr. Christoph von der Malsburg
Frankfurt Institute for Advanced Studies
Max-von-Laue-Str. 1
60438 Frankfurt a. M.
Germany
E-mail: malsburg@fias.uni-frankfurt.de

ISBN 978-3-642-00188-8 e-ISBN 978-3-642-00189-5

DOI 10.1007/978-3-642-00189-5

Library of Congress Control Number: 2010921797

© 2010 Springer-Verlag Berlin Heidelberg

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: Camera-ready by author, data conversion by Markus Richter, Heidelberg

Cover Design: eStudioCalamar, Figueres/Berlin

Printed in acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Contents

Prologue 1
0.1 Main Results 2
0.2 Prehistory of Our Project 6
0.3 Acknowledgement 7

1 The Difficulty of Modelling Artificial Brains 11
1.1 McCullogh-Pitts Model 12
1.2 Learning Nets 12
1.3 Spiking Neurons 14
1.4 Architecture of Vision 15
1.5 The Steps of the Construction Process 17
1.6 Summary 18

2 Information Processing in Nets with Constant Synapses . . . 21
2.1 Generic Signal Equations for Pulse Neurons and Synapses 22
2.2 Partitions and Their Time Development 23
2.3 Experiments with Constant Synapses 25
2.4 Entropy and Transfer Function of a Net 31
2.5 Operating Range of a Net 33
2.6 Pulse Rates 33
2.7 Resolution and Net Size 36
2.8 Application Potential 40
2.9 Limited Simulation Time 44
2.10 Summary 44

3 Theory of Nets with Constant or Dynamic Synapses 47
3.1 Derivation of the Signal Energy 47
3.2 Temporal Mean and Spatial Mean 51
3.3 Determination of the Frequency Distribution 53
3.4 Summary 58

4 Macro-Dynamics of Nets with Constant Synapses 59
4.1 Known Synapses 59
4.2 Known Distribution of Synapses 63
4.3 Agreement of Theory with Experiment 66
4.4 Lack of Correlation 79
4.5 Determining the Signal Energy and Entropy by Pulse Rates .. 79
4.6 Summary 81

5 Information Processing with Dynamic Synapses 83
5.1 The Types of Solutions of Synaptic Equations 84
5.2 Synchronisation of Neurons 85
5.3 Segmentation per Synchronisation 91
5.4 Calculation of Pulse Differences and Sums 93
5.5 Simple Applications 96
5.6 Time Coding and Correlation 100
5.7 Entropy and State Space 101
5.8 Preliminary Considerations on the Statistics of Synchronisation 103
5.9 Summary 103

6 Nets for Feature Detection 105
6.1 Overview of Visual System 107
6.2 Simple Cells 108
6.3 Creation of Detector Profiles for Gabor Wavelets 111
6.4 Experimental Check 115
6.5 Summary 117

7 Nets for Feature Recognition 119
7.1 Principles of Object Recognition 121
7.2 Net Architecture for Robust Feature Recognition 123
7.3 Feature Recogniser 126
7.4 Selectivity 128
7.5 Orthogonality of Rotation 131
7.6 Invariance of Function as to Brightness 132
7.7 Invariance of Function as to Form and Mimic 132
7.8 Generating Object Components through Binding of Features .. 136
7.9 Summary 138

8 Nets for Robust Head Detection 141
8.1 Results of Head Detection 142
8.2 Next Steps 146
8.3 Summary 148

9 Extensions of the Vision Architecture 149
9.1 Distance-Invariant Feature Pyramid 149
9.2 The Inner Screen 161
9.3 Summary 170

10 Look-out 173
10.1 Data Format of the Brain 173
10.2 Self-Organisation 174
10.3 Learning 176
10.4 Invariant Object Recognition 177
10.5 Structured Memory Domains 179
10.6 Summary 181

11 Preliminary Considerations on the Microelectronic Implementation 183
11.1 Equivalent Representations 183
11.2 Microelectronic Implementations 184
11.3 Models of Neurons and Synapses 186

12 Elementary Circuits for Neurons, Synapses, and Photosensors 195
12.1 Neuron 195
12.2 Adaptive Synapses 205
12.3 Photosensors 210
12.4 DA-Converters and Analogue Image Storage 230
12.5 Summary 231

13 Simulation of Microelectronic Neural Circuits and Systems 233
13.1 Modelling of Neurons and Synapses 234
13.2 Results of Modelling 235
13.3 Notes on the Simulation Procedure 237
13.4 Summary 246

14 Architecture and Chip Design of the Feature Recognizer ..247
14.1 Chip Architecture of the Feature Recogniser 248
14.2 Interfaces for Test and Readout 249
14.3 Chip Design and Layout 252
14.4 Demonstrator and Measurement Results 252
14.5 Summary 254

15 Architecture and Chip Design of the Feature Detector 255
15.1 Digital Representation of the Feature Detection 255
15.2 VLSI Design of a Neural Processor and Router Circuit 257
15.3 Demonstration of the Feature Detection 263
15.4 Summary 267

16 3D Stacking Technology 269
16.1 Fundamentals of 3D Stacking 271
16.2 Processing Steps for 3D Stacking 273
16.3 Assembly 292
16.4 Electrical Properties of 3D Interconnections 296
16.5 Summary 298

17 Architecture of First Generation Vision Cube 299
17.1 Imager Chip 299
17.2 AER Chip 300
17.3 Size of the NPU Array and Memory Requirements 304
17.4 Chip for Feature Detection 310
17.5 Chip for the Feature Recogniser 312
17.6 Summary 321

18 End 325

19 Appendix 327
19.1 Simulator for Chapters 3, 4 327
19.2 Axon Model by Hodgkin and Huxley 334
19.3 Basic Transistor Circuits 338
19.4 Optical Generation 344

References 347

Index 355

Prologue

Our brain has an estimated processing power of 10^{15} operations per second (counting one operation per synapse and second). It is thus a very serious computer. With its flexible, autonomous action and its ability to acquire, structure, index and coordinate world knowledge, our brain is vastly superior to any artificial system. In addition, it shouldn't be forgotten that the human brain is the source of all formal systems and algorithms. And although science has amassed tremendous amounts of knowledge about the nervous system, we still don't understand how it functions.

For more than a hundred years the human and animal brains have been the subject of research. Tissue slices are being produced by the millions, close to a hundred electrodes are put into the brain simultaneously, innumerable experiments are being performed concerning the electro-chemical and molecular processes in synapses and neurons, pictures of parts of the brain are produced with a variety of techniques, and hosts of behavioural experiments are being done. All of this research puts constraints on the putative mode of action of the brain. However, solid theory and quantitative modelling is restricted essentially to the level of single neurons and synapses [1]. Nets of neurons that perform certain functions of information processing, on the other hand, have been elucidated, if at all, close to sensors (e.g., receptive fields of retinal cells) [2–4]. Therefore, the majority of studies have mostly contributed to phenomenological description, accompanied by hypotheses concerning mechanisms of action. Consequently, we are far from any accepted description of the principles of information processing in the brain and from the reconstruction of its capabilities.

In this situation we have set out to determine how information is represented by the structure of elementary neural nets and their firing patterns. For this, we are exclusively using models of neurons and synapses that are simple enough to be simulated within a reasonable time and complex enough to solve non-trivial recognition problems. It is not our intention simply to add yet another puzzle piece to the state of the „neural art" described above, but

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 1–9, 2010.
© Springer-Verlag Berlin Heidelberg 2010

we aim to construct a little artificial brain able to complete an ambitious task, by appropriately combining those elementary information processing nets.

In order to prepare the reader, we give here an overview of our approach and of the most important results.

0.1 Main Results

Generic Recognition Chain in Neural Architecture

Limiting ourselves to vision, the main result we have achieved is: there are modules for the detection and recognition of features, a net for binding various features into one object component or of object components into an object, which in appropriate couplings, achieve a robust detection of human heads (for mimic, pose, rotation, and illumination invariance cf. chapter 8; for distance invariance cf. chapter 9). In this way, we have a minimal but complete chain of recognition modules; we refer to it below as „recognition chain". We have a special approach in that we let the whole set of functions for detection and recognition of features and for object binding be realised on the basis of a single model for neurons and synapses without any addition in terms of learning algorithms or something similar. We suspect that our approach is universal, and that a similar structure of modules and their coupling can serve as a basis for arbitrary recognition tasks. The input image could, for instance, be a visual scene containing various objects, in which case features would be defined by the presence of specific objects in the scene.

The universal reusability of the neural modules used becomes also apparent with the reconstruction of the retina image: the „spectral" decomposition of the retina image by means of feature filters is undone by exactly the same structure which was used to extract features from the retina image (chapter 9).

Representation of Information by Entropy and Correlations

A second important result concerns the quantitative characterisation of information processing in nets of pulsing neurons in terms of entropy and correlations. We take as a net's state space the set of all firing patterns, that is *partitions* of neurons into receiving and sending neurons, and consider the distribution of frequencies of occurrence of all possible partitions. By computing the entropy of this distribution, we capture the basic function of the net, which stands in a unique relationship to the firing rates of the neurons of the net. On the other hand, the correlation reflects the temporal relations of the neural firing activity in the net, which thus determines the structure and size of the state space. If, for instance, eight out of ten neurons fire in a specific and constant temporal relationship, then the distribution and the entropy are reduced to functions of the eight-neuron cluster and the two remaining neurons.

As we will see in chapter 5 and 7, synchronisation (simultaneous firing as a specific form of temporal relation) occurs when the expected features of an object to be recognized are present in the input image. The degree of synchronisation is evaluated by a dedicated net and is represented in the form of a firing rate. The encoding of information thus has the form of the two quantities entropy and correlation, both represented by firing rates. The old controversy between neural rate or spike time codes would thus be resolved, both being necessary for the transmission of information.

Pattern Recognition with Dynamic Synapses in the Example of Human Heads

The third important result is the application of such nets. It turns out that nets with constant synaptic weights, or synapses with fixed temporal behaviour, are not capable of a realistic pattern recognition that is robust against initial conditions, signal noise and the finite precision of synaptic weights. In such standard nets, signal correlations are typically very small, going to zero in the limit of large numbers of neurons, so that the state space cannot contain clusters or other patterns. Such patterns arise only in nets with dynamic synapses and can be exploited for recognition tasks. We describe such dynamic synapses, whose temporal behavior depends on incoming spikes and the membrane potential of the target neuron, by a simple differential equation that we developed. The functional power of these dynamic synapses is essentially due to the correlations and decorrelations that are generated by excitatory synapses (positive sign of the interaction term of the synapse equation) resp. inhibitory synapses (negative sign). Dynamic synapses play a central role at all levels of the recognition chain, especially feature detection and object binding. Important here is a small number of subnets with a special connectivity structure (chapter 5-9). In this way, information is represented in the whole net by the entropy-correlation tuple. It manifests itself in the image as a set of spatially distributed features, which selectively combine to represent objects or special compositions of objects in an image, etc. In this way we can connect the commonly accepted concept of information with a well-defined information concept using the entropy-correlation tuple.

Theory of Neural Network Dynamics

As our fourth important result we present the formulation of a theory that can predict quantities like entropy and firing rate as a function of input signals. Of central importance here is the definition of a signal energy, which may be identified with charges that collect within the neural membrane, and with currents passing through synapses. With the help of this signal energy it is possible, on the one hand, to recover the original signal equations for synapses and neurons, and on the other hand, to compute entropy and correlations in a net. In the special case of constant synaptic weight or synapses with rigid

temporal behavior we are able to show (chapter 3-4) that measured and computed entropy values differ by less than 3%. Admittedly, one particular result for dynamic synapses is still outstanding, although we can define the formal framework of signal energy and partial differential equation for the partition function. In the special case of synchronous neural nets with observable state space, an integral solution is available, however. With this we now have a theory with which to approach the informational processes in a brain.

Both experiment and theory require considerable processing power for the simulation. As this is limited, the size of the nets we could deal with in this book is restricted. The experimental determination of firing rates is possible up to nets of about 3000 fully connected neurons on a PC, and estimated entropy values can be sampled for 100 fully connected neurons. Exact entropy values can only be obtained on a PC for less than 30 fully connected neurons. Unfortunately, patience has its limits, and in this book we never consider more than 11 fully connected neurons when exact entropy values are at issue. Even with all the anticipated progress in the coming ten years, computing power will only suffice for the simulation and storage of typical input-output relations for small modules. The combination of modules into a full net will always be restricted to a limited number of test images. Theory therefore provides only a design model and helps us to get an idea of the action of the whole net, but to realise more serious applications, it will be necessary to build specific electronic realisations of the modules. These will be more powerful in magnitude than PCs, and will permit us to process and test a large number of real images in real time (chapter 17).

Neurocomputer of the Second Generation

As our fifth important result we therefore present a modular architecture together with its realisation in terms of CMOS technology. With this it will be possible to realise the recognition chain and in principle arbitrarily large nets. The corresponding chapters deal with circuit technology for the realisation of image sensors, neurons, synapses as well as modules for the detection and recognition of features and objects (chapter 12 - chapter 15). It emerges that the experimental behavior of the chip modules agrees closely with simulations and theoretical treatment. As the individual chip modules can easily be a thousand times faster than biological nets and as they are universally applicable, they can be temporally multiplexed to emulate a much larger net. In this way, complete feature pyramids can be realised, for instance. This requires, however, an appropriate architectural concept as a basis for the reading, storing and multiplexing of the signal values of neurons and synapses. This in turn necessitates the addition of a classical computer system to control and administer the resources on the one hand, and the propagation of the neural signals on the other. As distinct from the first generation of neurocomputers, which were either analogue or digital, the second generation will therefore have to exploit both computing styles.

Because of the multiplicity of necessary modules - only a few will differ from each other, due to the above-mentioned universal applicability of the recognition chain - it was necessary to develop a new technology of assembly: individual silicon slices are stacked upon each other and vertical vias complement the connectivity of the individual slices. In this way we have been able to connect six layers with a mutual distance of 14 μm each, although this stack was a mere dummy to test the quality of the connections and the processing yield (chapter 16). The next step will have to integrate the existing chips for feature recognition and object binding into a 3D stack. A proposal for this is contained in chapter 17 : a complete feature cascade of 8 orientations and 5 resolution levels has been realised in a vision cube comprising 59 chips of about 60mm^2 area each, stacked with a pitch of 14 μm to 1 mm height, as the basis for variable-distance recognition of 1000 objects.

From the Programmable to the Self-Organized Computer

Although in this book we can present only a first generation artificial brain and believe many more generations will have to follow to reach the full power of the human brain, we nevertheless see a new era of computation on the horizon. There were times when computers, with their precision, reliability and blinding speed, were considered to be superior to the wet matter of our brain, as a jet plane to a sparrow. That these times are over is clear, if we only consider the results described in this book, as well as the fact that digital systems that are inspired by formal logic and controlled algorithmically - today's computers - are hitting complexity limits. A paradigm change is in the air: from the externally organised to the self-organised computer [5].

Moore's law is both an opportunity and a challenge. On the one hand, growth in available computing power brings the emulation of the brain within reach, on the other hand, we will reach this El Dorado only by developing completely new principles of organisation. One aspect of this challenge is the organisation of structures such as the ones described in this book in their variety and massive parallelity. Another lies in the development of a new concept of „computing" that aims not so much at computing in the narrow sense but rather at the coordination of massive pools of data, at overcoming the division of labor between man and machine that is the foundation of the algorithmic concept. It also aims to implement the creative infrastructure in the machine that up until now has almost exclusively been residing in the programmers' brains.

The midwives of computer science were mathematical logic and formal systems. In the future we will have to get ready for dynamic systems that are to be described with probability methods, statistics, and non-linear differential equations. In particular it will, on the one hand, be necessary to develop a phenomenology of the data structures that are relevant for organised systems, on the other, it will be necessary to develop theoretical tools beyond those presented in this book, tools with which it will be possible to mediate between

macroscopic data structures and microscopic signal equations for neurons and synapses. In neural language this means asking the questions : What is the structure of mental objects, and How are these connected to the action of nerve cells?

All of this amounts to a dual strategy with two complementary developments: Realisation of a technology of massively parallel „organic“ computers after the model of the brain, and the development of a self-organising data-and-process architecture. Attempts of the eighties to build big processor arrays have failed mainly due to lack of an architectural concept of information processing. This latter will require fundamental cultural change.

0.2 Prehistory of Our Project

Before we start, here are a few words about the origins of our project. It arose in the division of „VLSI System Integration“ of the Corporate Technology of Siemens AG and built upon the project „Neurocomputer“ that ended in 1994 and was led by Prof. Ramacher. The neurocomputer *SYNAPSE* („SYnthezising Neural Algorithms on a Parallel Systolic Engine“) was only able to realise analogue neural models based on signal averages and sigmoid neural input-output functions. Unfortunately, it turned out after a few years that the neural architectures in those days were not powerful enough for applications such as vision. In particular, the neural nets at this level of development were only peripheral elements in the context of the architecture of applications.

With the support of Prof. Werrmann of the Federal Ministry of Education and Research (BMBF), Prof. Ramacher started the follow-up project „Electronic Eye“. In the year 2000, together with Profs. von Seelen, von der Malsburg (Ruhr-University Bochum), and Buhmann (University Bonn), this project generated a computing platform with the highest possible level of performance for prototypical applications such as automotive systems, surveillance in subway stations and video conferencing. The central component of this system was the „Vision Instruction Processor“, a highly integrated chip with a big parallel computing array and a large set of instructions for general computing, as well as more specialised instructions for image processing, image coding and decoding, image generation, and analogue neural nets. The whole system comprised a classical DSP architecture with an extended instruction set, so that general image processing algorithms as well as neural algorithms could be implemented. At the end of the project it was possible to demonstrate real-time applications such as a passing-car alert, face recognition, MPEG2 coding of TV images and two-dimensional image generation. The number of neural components in the whole image recognition system continued, however, to be small.

Not until the next project „VisionIC“, which was designed by Prof. Ramacher and supported by Prof. Werrmann (BMBF), was a component directed

at research on firing neurons, dynamic synapses and their information processing capabilities, thus realising a step towards a pure neural architecture. Prof. Ramacher and co-workers (since April 1999 at the Corporate Research Division of Infineon Technologies AG, formerly the semiconductor division of Siemens) now started to work on the development of fundamental experiments and the theory of information processing in nets of firing neurons and on steps towards the architecture of the recognition chain described above. Meanwhile, the realisation of the corresponding silicon chips with firing neurons and dynamic synapses was entrusted to Prof. Schüffny and co-workers at the Technical University Dresden and the realisation of a 3D stack of chips with vertical vias for sufficient wiring in large artificial brains to Profs. Kohn and Pfleiderer (University of Ulm).

In the main part of the VisionIC project, Profs. von der Malsburg, von Seelen and Buhmann continued their efforts to combine classical computer vision and neural architecture in an advantageous way. The neural component of the project profited very much from the tension between fundamental and application-oriented research. On the one hand, the applied scientists were not burdened by having to realise every information processing step in neural detail, allowing them to make faster progress. On the other hand, they thus demonstrated state-of-the art research, that is, the limits of robust object recognition, limits that after forty years of computer vision research are still evident. At the same time it was fascinating to see that at least when limiting the general object recognition problem to human heads at a fixed distance, the neural-inspired architecture portion could dominate in the whole system. In particular, the Dynamic Link Architecture of Prof. von der Malsburg [6] was an ongoing inspiration for the project component team to attempt its realisations on the basis of firing neurons and dynamic synapses.

The results of this component of the project „VisionIC“, which ran from 2001 to 2004 with the support of the BMBF, and of subsequent workings are described in this book. We are conscious of the fact that the contents of this book constitute only first, small steps and that we cannot claim that we have completely deciphered the architecture of vision. We hope, however, that in the end the reader can agree with us that we have found the start of a journey that will lead to new insights.

Beyond that, it is a goal of this book to make available the outline of a new discipline that incorporates experiment, theory and implementation with equal weight and combines them in the construction of artificial brains.

0.3 Acknowledgement

The program of the editors would not have been carried out and this book would not have been written without the help of Profs. Kohn and Pfleiderer (University Ulm), Rückert (University Paderborn), Schüffny (Technical University Dresden) and without the help of their doctoral and diploma stu-

dents. Likewise it would not have been written without the colleagues at Siemens resp. Infineon (abbreviated as S and IFX) who have collaborated in our projects.

Chapters 1 and 10 were written by Chistoph von der Malsburg, chapters 2-9,11 by Ulrich Ramacher, the material of chapters 12-15 was prepared by Jens-Uwe Schlüssler, and chapters 16-17 were written in whole or in part by Arne Heittmann. The tuning and final editing of all chapters was done by Ulrich Ramacher. Individual chapters have profited from the support of the following persons:

- Chap. 2: Helge Ludwig (IFX) has developed and simulated the muscle control. All other simulations have been performed by Dr. Arne Heittmann (S/IFX);
- Chap. 3: Numerous conversations with Dr. Wieland Jass (S) helped in the foundation of the theory of neural information processing;
- Chap. 4: The efficient programming of the theory and the performance of all simulations have been carried out by Jens Harnisch (S/IFX);
- Chap. 5-8: All neural nets have been programmed by Arne Heittmann, who in addition has created the basic design for the pulse rate difference circuit in Chap. 5 and for the Gabor wavelets in Chap. 6;
- Chap. 7-8: Prof. Rückert, Dr. Ulf Witkowski and Tim Kaulmann have used the parallel computer of Paderborn University to simulate some complex nets for feature detection and recognition;
- Chap. 9: all simulations regarding the distance-invariant feature cascade and the reconstruction of the retina image were performed by Arne Heittmann;
- Chap. 12-15: Doctoral students Jörg Schreiter, Chistian Mayr, Holger Eisenreich, Stefan Henker, Daniel Matolin and Jens Döge, together with Jens-Uwe Schlüssler and Prof. Rene Schüffny (both of TU Dresden) and Arne Heittman, have developed the circuits and chips for feature recognition and detection;
- Chap. 16: Alexander Kaiser, Andreas Munding and Peter Benkart under the guidance of Prof. Kohn and Markus Bschorr under the guidance of Prof. Pfleiderer have successfully developed and tested the individual processing steps of the 3D integration; essential contributions to the 3D stacking technology presented here are due to Dr. Holger Hübner (Infineon) and his work in the context of SoLID.
- Chap. 17: the costs of the vision cube in terms of gates, memory, and chips were determined by Arne Heittmann (IFX).

To all of these persons we would like to express our heart-felt gratitude - we have been a wonderful team!

We finally would like to mention professor emeritus Werner von Seelen (formerly Ruhr-University Bochum), who has actively followed the project for over ten years and for whose constructive criticism we are infinitely grateful.

At last a great thanks to my friend Fay Millikan from Hobart, Tasmania, who undertook to correct our English translation, and to Dr. Ditzinger, Heather King and colleagues from Springer-Verlag for their ideal mentoring and understanding.

Munich, July 2009 *Ulrich Ramacher*
Christoph von der Malsburg

1

The Difficulty of Modelling Artificial Brains

Christoph von der Malsburg and Ulrich Ramacher

By far the most successful version of information technology is exhibited by the brain. Animal and human nervous systems are flexible in dealing with the unexpected. They adapt, learn, evolve and integrate well into social networks, have an innate tendency to organise, are energy-efficient and massively parallel, and above all they act in total autonomy. In all these respects, nervous systems surpass technology, which no doubt should profit from this example. Unfortunately our knowledge of structure and function of the nervous system, though extensive, is far from being sufficient to serve as a direct manual for reconstruction. Nevertheless, our accumulated knowledge gives us boundary conditions that may already now be distinct enough to guide functional reengineering in promising ways. From experience, the neurosciences will profit in return, for the attempt of a technical emulation is a very exact test to separate the sound ideas of brain function from the faulty ones.

Due to its digital construction, the computer is a deterministic, logical, static entity under algorithmic control. By contrast, the brain is a non-digital, non-deterministic dynamic system with continuous, noisy computing elements. Its dynamic trajectory is not the result of external programming but of self-organisation of interacting computing elements under the influence of input signals.

While the computer is guided by the insight of the programmer, the brain has an innate tendency to organise. To understand this, it is necessary to apply methods similar to those used in physics, which exploit the basic laws of local particle interactions to describe the macroscopic phenomena generated by masses of particles. Applied to the computing elements of the brain, this means to derive the information processing functions of the diverse brain areas and their submodules from the basic laws of interaction of action potentials, synapses and neurons. This task has occupied a number of disciplines again and again and it has proceeded in several phases.

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 11–19, 2010.
© Springer-Verlag Berlin Heidelberg 2010

1.1 McCullogh-Pitts Model

The neurons in our nervous system are complex organisms, direct if distant heirs to our autonomous, single-celled ancestors. Like all other cells of our body, almost without exception, every neuron carries a full genome of about one GByte. Each of our neurons is probably comparable or even superior in its complexity to any current computer, including the operating system. Their evolved complexity makes them capable of autonomous organisation. This capability expresses itself already in the regulation of the neuron's duty cycle [7, 8], an important aspect of energy optimisation [9], and especially also in the balancing of the load of information processing over neurons.

The brain, as a three-dimensional entity, is constructed inside-out by its building elements, neurons and other cells, in a genetically controlled growth process. One may see the neuron as the equivalent of a microprocessor, and in this sense our brain with its estimated 10^{12} neurons and 10^{15} connections is a massively parallel, intensely interconnected system. The individual building elements - cells, axons, synapses, dendritic branches - are actively regulated elements with multiple nested control networks. While modelling the brain, the first great challenge therefore is the reduction of the complexity of the biological neuron, including dendrites and synapses, necessitating the formulation of an abstract model. A system of partial differential equations that renders the exchange processes of the diverse membrane channels of a neuron [10] is not useful if the information-processing of a large number of interconnected neurons is to be studied. On the other hand, the model should not be too simple.

In the year 1943 McCulloch and Pitts proposed a formalisation of neural brain cells [11], involving discrete time and binary signals, just as in today's digital computer. The central point of the publication was that an arbitrary logical function could be realised in this way. This implies also that a universal Turing machine (except for the infinity of its tape), and in consequence every imaginable symbol manipulation procedure, can be implemented in McCulloch-Pitts neurons. This seemed to break down the impenetrable wall between mind and matter. Unfortunately this picture is too simple. The question remains regarding the origin of the wiring of the system, and the origin of the functions and procedures that are to be implemented. Furthermore, due to their rigid synaptic connections, McCulloch-Pitts networks cannot serve as a basis for self-organisation.

1.2 Learning Nets

Since Turing or McCulloch and Pitts the modelling of the brain and its function have been vigorously developed further, both from the algorithmic perspective in the framework of Artificial Intelligence, and from the neural perspective (that is, with emphasis on self-organisation), with models such as

Associative Memory [12–14] or layered neural networks [15–17], the latter under names such as connectionism or artificial neural networks (ANN). Behind these names are a multitude of types of networks [18, 19], which, however, may not differ always with respect to the function they perform.

A second class of neuron models that were very popular in this era, although unable to fulfil the expectations with regard to their explanatory vigour, disregards the original discrete firing of neurons, is based on mean firing rates, and uses a neuron model that transforms analogue input signals by means of a sigmoid function into an analogue output signal.

Then and now the neuronal perspective puts the main emphasis on the generation of the structures of interest by (statistical) learning, which often is realised as matching the actual output to a teaching signal, by reduction of an error signal. A central role in this game is synaptic plasticity: the plastic modification of coupling strengths between cells under the control of locally given signals, strengthening those synapses whose transmitted action potentials have been successful in influencing or predicting the signal of the target cell.

Unfortunately this doesn't solve the problem of learning from the natural environment, and shows up the limitations of the ANN most conclusively. A neural network only learns efficiently if the input patterns which are presented contain not more than a couple of 100 bits of information and if all input patterns belong to the same context [20]. If these limits are not observed, the number of necessary training patterns, and with it the training time, grow to astronomic dimensions. The reason is that with the increasing size of input patterns and of the system, the significant neural connections that are to be found (that is, those that express essential structural relations) are getting submerged in an ever bigger sea of insignificant connections, such that by statistical means they can no longer be identified. The more profound reason for this difficulty of the ANN lies in their inability to express binding between neurons, let alone to actively decide that issue in a given situation. This is the so-called binding problem [21].

The 80es in the last century saw an „assault at the brain" on the basis of this learned interconnection between analogue neurons [22]. All in all, we can say that the ANN have led us to a multi-facetted methodology of statistical estimation, which found broad application in the industrial context, but the aspirations of 20 years ago - to understand the brain and to create an information technology based on self-organisation - have not been realised up to the present. Specifically, artificial neural nets are fit only for a narrow spectrum of tasks (e.g., command and control [23]), and they are not appropriate for the construction of whole complex systems but only for small subsystems in an algorithmic harness. Thus, concerning applications like image and speech perception, we can speak of a „neural gap", which persists to today.

1.3 Spiking Neurons

As pointed out, the ANN of the 80s and 90s are not apt to render the cognitive architecture of the brain. Evidently it is necessary to get our models of neurons and synapses to approach biological reality more closely. Instead of average spike rates, spikes[1] or whole spike sequences and the neuron models that produce them are considered, and at the same time learning algorithms or variants of the Hebb rule are applied to spiking neurons and slowly varying synapses [24]

By taking into account the temporal signal structure, it is possible to use temporal correlations between neurons to encode information to complement spike rates. *Temporal binding* in the brain, and with it the importance of temporal signal correlations in the specific form of synchronously spiking neurons are experimentally well documented [25]. Unclear is, however, how temporal correlations are generated, how they are evaluated and how they influence the architecture of information processing modules in the brain.

There are many hypothetical answers to these questions. For example, in the construction of our functional models (outlined in chapter 5-9) we let ourselves be guided by a hierarchical schema of neural action. This schema has the form of a tree graph that is composed of nodes and rapidly modifiable dynamic links. Nodes may represent single neurons, clusters of neurons, or sets of clusters. Links communicate signals between nodes. Each node has a large number of links to other nodes with which it has been associated over the course of time. In a given situation, however, only a small subset of these links is relevant. Graphs are therefore to be subject to a process of dynamic link modification by which all links that are currently not relevant are to be switched off. We assume that links are realised by rapidly modifiable synaptic connections, which in their turn are controlled by the correlations between the signals of the neurons that they connect. With these hypotheses and abstractions it has been possible to bring applications of human face recognition to industrial application [26].

It is our purpose here to test graph dynamics in concrete neural terms. Therefore, we constrain ourselves in this book to concrete neural models based exclusively on neurons, synapses and their local, pulse-based interactions, and to refrain from learning algorithms or similarly obscure processes at this stage of information processing. More concretely, we take a simple integrate-and-fire model neuron and a simple ordinary differential equation for the synapse as starting point for the construction. Nevertheless, the nets we describe realise a complete „object recognition chain“ and perform the non-trivial tasks of robust detection and recognition of features, binding of features into objects and classification of objects, as will be explained in chapter 6-9. The results seem to confirm that we have struck the right balance between simplicity of

[1] In subsequent chapters we will consider idealized spikes, namely rectangular pulses.

structure and performance of information processing. Although the realisation in the forementioned concrete sense has only begun graph dynamics appears to be an indispensable constitutent of the brain's architecture.

1.4 Architecture of Vision

A successful cognitive architecture can only be developed in the framework of a practical, paradigmatic application, either for purposes of information technology or for neural modelling. Correspondingly, it is important to select a brain function with generic features and to model and understand it in detail. Vision plays for us humans a central role, and there are extensive research results in this field, therefore this seems ideal as a paradigmatic application. In addition, a successful emulation of the biological vision process is of great scientific and economic value.

Vision, admittedly, is a very complex process that involves many interacting sub-processes, and about a third of our brain is occupied by it. We concentrate in this book on merely two sub-processes - image segmentation and object recognition. In this we can build on several decades of research in neurophysiology, psychology and technical image processing.

A central role is played by the extraction of local image features, for which Gabor-type wavelets have proven to be particularly relevant [2, 27]. Also important is knowledge about the hierarchy of the cortical areas [28] which emphasize different image aspects, such as local form, color or motion, and which realise a progression towards more and more invariant representations, see for example [29].

The human vision process is of sequential nature, selective attention being directed towards one structural element after another (coherent object, object part or similar), the whole system coordinating all of its resources to analyse and represent this structural element. An important aspect of this process is segmentation, that is, the separation of the structural element from the background. The assembly of all parts of a structural element into one whole is often cited as a canonical example of the binding problem [30, 31], and there are numerous models that propose to solve this problem by temporal signal correlations, see for example [32–35].

Paradigmatic for issues of organisation or architecture of systems is the problem of invariant recognition. The appearance of an object in the retinal image varies, and the system has the task of recognising the object independently of translation, scaling, rotation, deformation, partial occlusion, illumination or signal noise, and must represent the structure of the object in invariant form, so that other system components (speech, motor control etc.) can react undisturbed by image variation.

An early proposed solution for the problem of translation invariance [15] assumes four system layers. The first layer is the image domain, a two-dimensional manifold populated by point detectors. The second layer contains

variant features: neuronal elements that extract local patterns or „features“ (e.g., Gabor wavelets), so that each of these units is specific for a feature type and for a position. The third layer contains for each feature type only one unit, which is activated if at least one unit of this type is active in the second layer. In this way, the third layer represents an object as an unstructured list of features that apply to this object. The fourth layer contains units whose connections with the third layer are structured such that the units respond selectively to specific sets of units of the third layer, that is, to specific sets of invariant features. If the set of features realised in the second and third layer is sufficiently differentiated, the units of the fourth layer are selective for an object type. Systems of this general structure are usually called feature-based.

The feature-based approach is plagued by several fundamental problems. *First,* it is an unsolved, and possibly unsolvable, problem, to define a set of feature types such that no two objects that are to be distinguished contain the same set of features if in a different arrangement. This problem is aggravated if segmentation is not accurate and features of the background are also activated, and it is made even more difficult if invariance is required not only against translation but also against scaling and rotation, so that on the way from the second to the third layer, features of different size and orientation are identified with each other.

Given the complexity of our visual environment, the problem of the definition of a set of feature types has to be solved with the help of learning. This attempt leads, however, directly to a *second* problem of the feature-based approach, the hen-and-egg problem, that appropriate features can only be selected according to their contribution to successful object discrimination while object discrimination without appropriate features is not possible. A *third* problem with the feature-based approach is the lack of structure of invariant object representation in the third (let alone the fourth) layer, which makes it impossible for the rest of the system to perceive the relative position of the features: on the way from the second to the third layer , the baby - object structure - is thrown out with the bath water - object position. Numerous attempts have been made to solve the problems of the feature-based approach. One proposal, Fukushima's neocognitron [36] (revived in [37]), is based on the introduction of a multiplicity of layers, which contain a progression of feature types that, at the same time, become gradually more complex and more invariant. This proposal may solve the first of the cited problems, but not the second and third.

We will show in later chapters how these problems of the feature-based approach can be solved in a natural fashion with the help of dynamic links, which make it possible to transfer the spatial relations between features, which are still explicitly represented in the second layer of Rosenblatt's four-layer perception, to the third layer, so that they can also be explicitly represented there. The respective construction steps are summarised in the next section.

1.5 The Steps of the Construction Process

Universal Feature Detection

It has been shown that Gabor wavelets approximate the response rates of the so-called simple cells in the primary visual cortex V1 with great accuracy [2]. We present a real-time 6-layer network architecture that is able to reproduce the pulse rates of simple cells very precisely. This network is composed of constant as well as dynamic synapses and contains in the last, the sixth, layer only a single neuron. In this, the feature profile is not implemented by a weighting of all pixels but by coupling the inputs neurons of the net to selected pixels which by their locations define the shape of the receptive field. There is an impressive agreement between Gabor wavelets realised in this way with those defined mathematically. By the way, this network could realise arbitrary feature types besides Gabor wavelets.

We imagine the various types of Gabor wavelets as arranged in a stack of planes connected to the retina. Input images generate patterns of neural spikes in this stack, from which in the next step invariant subsets are selected.

Robust Feature Recognition

We assume that objects, say a human head, are seen from a distance and position such that their image fills the input layer. In that case, relevant features, referring, for instance, to the eyebrow, appear within a given region, which we designate as a „zone“. Each feature detector neuron has a one-way connection to a feature recognition neuron, whereas the recognition neurons are connected to each other, in general in a scheme that lies somewhere between all-to-all and next-neighbor connections. There usually are as many recognition neurons as detector neurons for a given zone. The recognition neurons activated by a feature synchronise with each other, independently of the position and form of the feature and independently of the strength of illumination (chapter 8) as well as independently from the distance (chapter 9).

Robust Object Detection

Each object image in the appropriate retinal position is covered by a number of zones, in each of which one or several features may appear. In the latter case different features are combined into one object component. In general, the number of zones of an object grows with the number of objects to be distinguished.

In the next step we bind the components of an object, that is, of all the zones, together. Only if a sufficient number of Gabor wavelet modules are activated in the different zones by the characteristic features of the imaged object will the object detector respond. In this way, objects such as human heads can be detected even with varying expressions, poses, positions, illuminations and in various distances.

Reconstruction of the Retina Image

Since the feature detectors occupy at least $2 \cdot 4$ pixels and many of these cooperate in the process of recognising a contour - say, when we try to seize the stem of an apple - and since we can recall images from memory before our mind's eye, there must exist a process creating an exact, pixel-fine positioning of the objects at the retina, their components, and their contours. The simplest and most resource-saving solution is the construction of a 2-dimensional „inner screen", which redesigns the retina image from the responses of the feature detectors.

In the same way as the feature image was created by convolving the retina image with the Gabor wavelets, the reconstructed retina image is obtained by convolving the feature image with the local inverses of the Gabor wavelets. This implies the reuse of the neural architecture of the feature cascade, only the distribution of sampling input neurons of the net must be changed accordingly (chapter 9).

1.6 Summary

In contrast to Rosenblatt's proposal, the availability of the very fast neurosynaptic binding mechanism presented in chapter 5 makes possible a correspondence-based solution to the invariant recognition problem. In our approach, the spatial arrangement of features, which in the second (feature detection) layer is explicitly represented, is preserved on the way to the third (feature recognition) layer. For each object to be recognised the third layer contains a two-dimensional network of feature-specific nodes that contains the features present in the object in the same arrangement as in the image. Other than what seems to be implied by the somewhat misleading terminology, the correspondence-based approach does employ features, but differs from the feature-based approach by explicitly retaining feature arrangement in the invariant representation. This turns the third layer into a complex model domain, with a set of feature networks for each object. The fourth layer binds sets of features together into objects. The recognition process must activate not only the correct network in the memory domain but also a whole system of point-to-point links between nodes of that network and the corresponding feature units of the second layer. These correspondences are realised with the help of dynamic synapses and cause the synchronisation of spike activity in the third layer.

The problems that we mentioned in the context of the feature-based approach are solved in a natural way by dynamic synapses in the correspondence-based system. In particular the first problem (re-arrangement ambiguity) and the third are solved by explicit representation of the spatial object structure. Also the second problem is decidedly reduced, in that the system can recognise objects unambiguously on the basis of a simple standard set of features (Gabor

wavelets, for instance), without requiring feature learning. The corresponding hen-and-egg problem thus disappears.

Chapters 5–8 address the particulars of the various steps of the construction process. Chapter 9 describes in all detail the neural architecture of the complete recognition chain for 1000 objects, including the feature cascade with 8 orientations and 5 resolution levels. The implementation in 3D technology is described in chapter 17.

In chapter 10 we give a perspective on further questions of invariant object and scene recognition.

2

Information Processing in Nets with Constant Synapses

Ulrich Ramacher and Arne Heittmann

In this book, we attempt to construct an artificial vision system similar to the human brain which can perform complete image recognition tasks, ie. those which comprise feature detection, binding of features to objects, and object recognition. Similarity we see implemented by the usage of neurons and synapses as simply as possible, plus the restriction to only these: rejecting any adjunct of algorithms additional to the ones defining neurons and synapses.

Accordingly, the signal behaviour of biological neurons and synapses is abstracted only as far as the said challenging image recognition tasks can be accomplished. More precisely, artificial neurons are considered which create a pulse when a threshold is reached, the membrane potential being set to zero for the duration of transmission. Subsequently the accumulation phase is started, with synaptically weighted external signals as well as internal pulses fed to the receiving neuron. Furthermore, synapses are supposed to change their strength relative to the incoming pulse and the present membrane potential.

The experiments as well as the theory are of outmost importance for our approach. By experiment we mean computer-based simulations of pulse neurons, synapses, and nets composed of those in which only the selected signal equations for neurons and synapses are used over and over. In theory the quantitative modelling of interrelationships is carried by the values measured in the experiments. Only after having brought basic experiments and theory into line, does the construction of feature detection and climbing of the recognition chain begin.

The goal of this chapter is, firstly, the definition of signal equations for pulse neurons and synapses. Secondly, we specialise in synapses with constant weights and, guided only by experiments, answer fully the question how information is represented and how it is processed.

This chapter does not give an introduction to neurons and synapses; for that purpose, see the relevant literature [38–40], and chapter 11.

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 21–45, 2010.
© Springer-Verlag Berlin Heidelberg 2010

2.1 Generic Signal Equations for Pulse Neurons and Synapses

Of the N neurons σ ones may transmit pulses at time t_0, and ε others may receive these pulses. Here: $\sigma + \varepsilon = N$, for $\sigma, \varepsilon = 0 \ldots N$, is always true. The form of the pulse can be shaped arbitrarily, for instance as combination of rising and falling exponential functions or as a sequence of those. However, for our image recognition tasks, rectangle signals are sufficient: all pulses last for t_d seconds (subsequently set to 1 ms) and have constant amplitude x; $x(t)$ is the time function of a pulse.

Further, let $E(t_0)$ designate the set of neurons receiving at time t_0, and $S(t_0)$ designate those neurons transmitting a pulse at this time. The initial state of the net at time t_0 is described by the elapsed transmission times $\Delta_s(t_0),\ s \in S(t_0)$, as well as by the accumulated membrane signals $a_e(t_0),\ e \in E(t_0)$, of the neurons receiving at this time. Finally, the instantaneous signal values $i_n(t_0),\ n = 1 \ldots N$, of the external input and of the synapses $W_{es}(t_0)$ and $W_{e0}(t_0)$, respectively, have to be recorded. Note that i_n may represent pulses as well. W_{es} and W_{e0}, resp., indicate synapses which mediate the pulse of the s-th neuron to the e-th neuron, and the e-th input signal to the e-th neuron, respectively.

As signal equations for receiving and transmitting of a neuron, there is the following arrangement[1]:

(1) Once a neuron(let it be the e-th) no longer transmits and as long as the threshold θ has not yet been reached,

$$0 \le a_e(t) := \int_{t_0}^{t} \left[\sum_{s \in S(t_0) \cap N_e} W_{es}(t')\, x_s(t') + W_{e0}(t')\, i_e(t') \right] dt' + a_e(t_0) \le \theta \tag{2.1a}$$

designates the accumulated signal value of the e-th neuron at time t. Here N_e is the set of neurons which are connected to the e-th neuron.

(2) Once a receiving neuron reaches the threshold or as long as a pulse-transmitting neuron has not stopped sending, it holds:

$$0 < x_s(t)\ , \qquad for \quad t_0 < t \le t_0 + t_d - \Delta(t_0)\ . \tag{2.1b}$$

We emphasize that no decay term for the accumulating neuron has been introduced so far, since we do not need it for the image recognition tasks considered in this book. More complex tasks may ask for such an extension, however. In order that any new information processing quality of an extension

[1] If neuron signals as well as thresholds are chosen dimensionless, synapses have the dimension Hz. Here care must be taken of the unit of time; in this book the ms is the unit, hence weights have the dimension of kHz. For sake of ease, no dimensions are given.

be brought to light clearly, we remove any term from our equations of neuron and synapse that is not yet needed.

We still need an equation describing the behaviour of the synapse. As local interactions are considered, the generic synaptic equation reads:

$$\begin{aligned} \frac{dW_{es}(t)}{dt} &= -\gamma \cdot W_{es}(t) + Q(a_e(t), x_s(t)) \ , \qquad \text{for } e \in E(t),\ s \in S(t) \ , \\ \frac{dW_{es}(t)}{dt} &= -\gamma \cdot W_{es}(t) \ , \qquad \text{for } e, s \in E(t) \ , \end{aligned} \tag{2.1c}$$

with Q a function (or even functional) of the pulses and the membrane signals, and γ being a decay constant. In later chapters, also synapses embedded in a dendrite are considered which, instead of the membrane potential, use a local dendrite potential (clearly, the equation (2.1c) can also be arranged to hold for input synapses).

Apparently, our synapses obey a generalised Hebb equation. Its exact form for successful operation will not be revealed until chapter 3.

2.2 Partitions and Their Time Development

When $a_e(t)$ reaches the threshold θ at time $t = t_e$, the e-th neuron will start to transmit a pulse. Consequently, the variable $a_e(t)$ disappears. Simultaneously, the partition into receiving and transmitting neurons has changed. Equally, the partition changes when some of the neurons stop sending: instead of the pulse signal, a membrane signal now exists which always starts at zero. This is in contrast to $a_e(t_0)$ which, being the initial value, is generally unequal to zero.

The point of time when a partition transitions into the next one can be determined uniquely (here external signals and synapses may embrace any time functions). Let

$$\tau_s(t_0, t) = t_0 + t_d - \Delta_s(t_0) - t \tag{2.2a}$$

be the current remaining life time of the s-th pulsing neuron at time t. Some of the pulsing neurons (say, the x-th one) will exhibit the smallest life time. If no receiving neuron reaches the threshold earlier, the life time of the first partition is determined by $\tau_x(t_0, t)$ becoming zero. As life time and final point of time, respectively, then result in:

$$\tau_0 = t_d - \Delta_x(t_0) \ \ , \ t_f(0) = t_0 + \tau_0 \ . \tag{2.2b}$$

Likewise, the currently estimated remaining life time $\tau_e(t_0, t)$ of the e-th receiving neuron is defined by

$$\tau_e(t_0, t) := \left[\frac{\theta - a_e(t_0)}{a_e(t) - a_e(t_0)} - 1 \right] \cdot (t - t_0) \ . \tag{2.2c}$$

Some of the receiving neurons will exhibit the smallest life time. However – depending on the dynamics of input and pulse signals – in the next moment a different neuron may get the smallest life time. If no transmitting neuron will decease before a receiving neuron has reached its threshold, the life time of the first partition is determined by the minimum (running over all life times) reaching zero at some time t. The receiving neuron with life time 0 will be designated by x again. Then, we obtain for the life time of the first partition the implicit expression:

$$\tau_0 = (t - t_0) \Leftrightarrow \theta - a_x(t) = 0 \ . \tag{2.2d}$$

Hence, in each partition and at each point in time, the minimum of all life times $\tau_e(t_0, t)$ and $\tau_s(t_0, t)$ has to be computed in order to determine the transition point to the subsequent partition. This procedure of calculating life time and final time point of a partition stays unaltered when an additional equation for the synaptic behaviour is considered.

In this way, we can obtain all partitions. It must be noted that no partition - except the one with no transmitting neurons - can live longer than a pulse. We can reasonably hypothesize that the average life time of a partition decreases with an increasing number of neurons in a net. Figure 2.1 confirms this conjecture.

Since the number of pulses received by each neuron would grow very rapidly with increasing numbers of neurons in a fully connected net, the

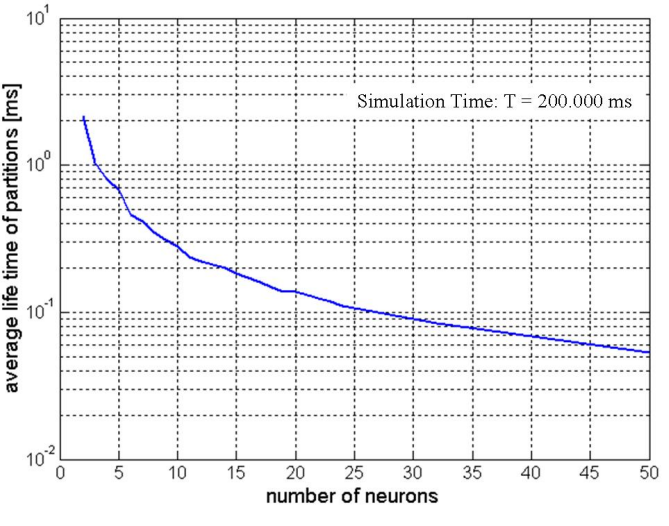


Fig. 2.1 Life time of partitions as function of number of neurons

synapses are scaled such that the product of neuron number N and average synaptic weight W is constant. Here, the average synaptic weight is computed from the interval $[0, 2W]$, and it is assumed that synapses are equally distributed therein.

Inspection of (2.1a) shows that the signal dynamics remains invariant if the quantities $W_{es}, W_{e0}, a_e(t_0), \theta$ are replaced by $W_{es}/\theta, W_{e0}/\theta, a_e(t_0)/\theta, 1$. In the sequel, we set the threshold to 1, ie. we get rid of a dispensable parameter[2].

Since an important class of neural nets is given by those in which synapses have constant or fixed time functions we start with the study of information processing for this case.

2.3 Experiments with Constant Synapses

We consider a fully connected net of 11 neurons each of which being fed an input signal via a synapse of value 0.1. Input signals have randomly chosen fixed values from the interval $[0, 1]$, and the synapses connecting neurons exhibit the mean value of 0.081 and mean variation of 34%. Duration of a pulse is chosen as $1\,ms$. When the net is activated, ie. the signal equations (2.1a–c) are computed for each neuron, a neuron's activity can be represented as a sequence of pulses (Figure 2.2). As a characterisation of the pulse activity can not easily be read off from Figure 2.2, the time distances between the

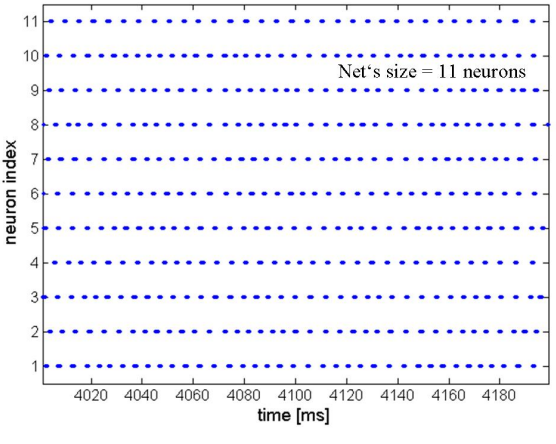


Fig. 2.2 Pulse activity of the 11 neurons as function of time, in the interval $[4000ms, 4200ms]$

[2] For reasons of computational stability and applicability, these parameters will have to be re-introduced in later chapters.

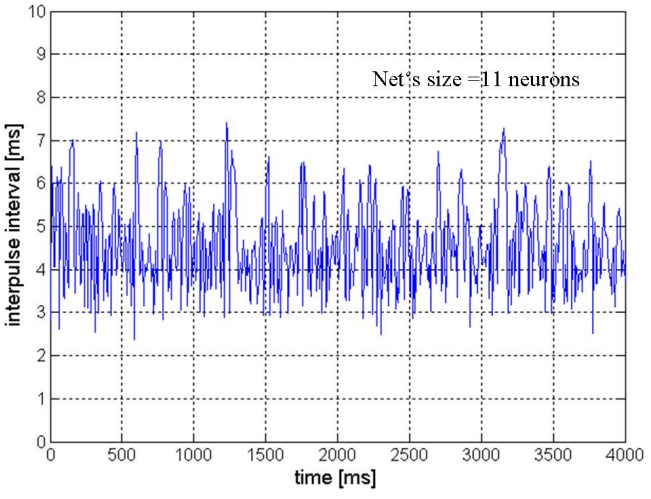


Fig. 2.3 ISI as function of time

pulses of a neuron, the so-called inter spike intervals ISI, are investigated next (Figure 2.3). Again, no order or regularity seem apparent.

Consequently, we record the histogram of the ISIs for each of the 11 neurons. After 100000 pulse events have elapsed for each neuron the histogram

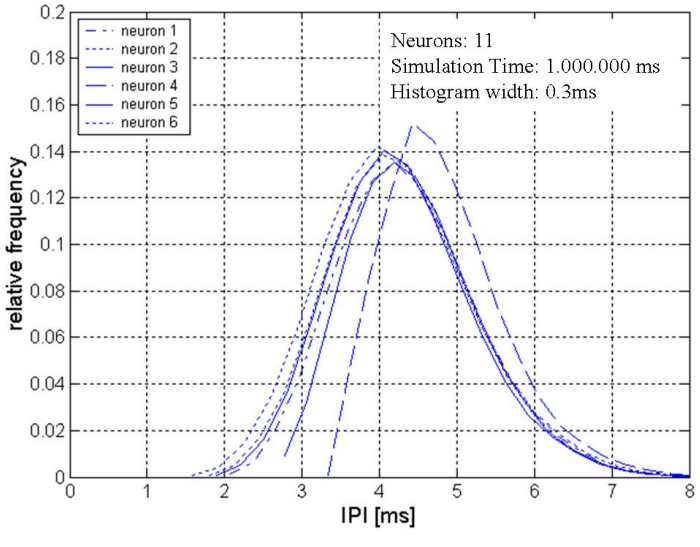


Fig. 2.4 Representative ISI histograms of some of the 11 neurons

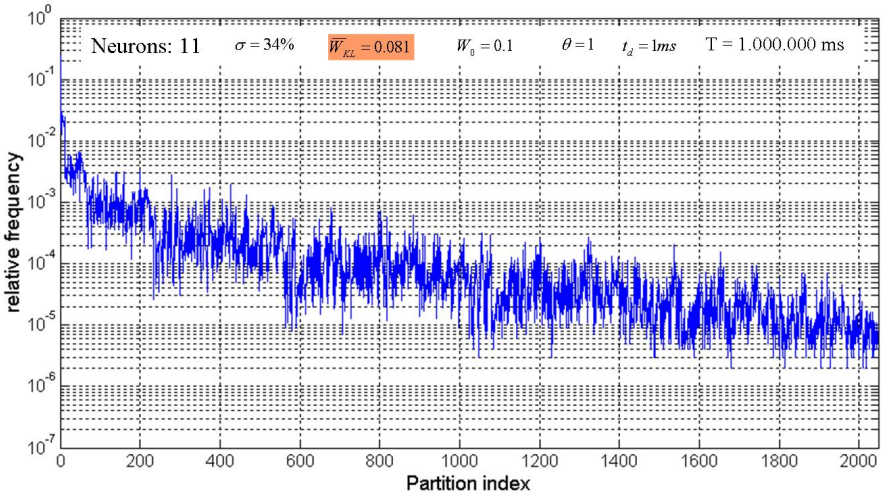


Fig. 2.5 Frequency distribution of partitions

becomes stationary (Figure 2.4). Obviously, the ISI histograms of the neurons don't signal any periodicity, but only broadly distributed values.
Finally, we are interested in the frequency of occurrence of a partition (Figure 2.5). Here, partitions are ordered in such a way that on the far left the partition with receiving neurons only is located, followed by the subset of partitions exhibiting 1 transmitting neuron. And so on. On the far right, the partition with transmitting neurons only is placed.[3].

The distribution curve in Figure 2.5 declines from left to right. This indicates that the net's neurons are predominantly receiving. When the mean value W of the synaptic weights between neurons is augmented the average number of transmitting neurons will grow rapidly.

Figure 2.6 shows the frequency distribution for the pair $(W = 0.18, w = 0.1)$ which exhibits an approximately horizontal gradient.

Further increase of W causes the average receiving time of a neuron to become smaller than the duration of a pulse. That is, neurons fire shortly after the last pulse has decayed. Consequently, the number of pulses contributing to the membrane potential of a neuron decreases strongly. In other words, the mixing process of pulses at a neuron's membrane from all over the network is highly incomplete, which causes some of the partitions to disappear. Thus, it is not lack of simulation time that causes this behaviour.

[3] This ordering is by no means distinguished, just as little as the ordering of partitions by value. In the latter case a smooth curve is obtained, but with larger networks there is little information available about the locations of partitions at the x-axis.

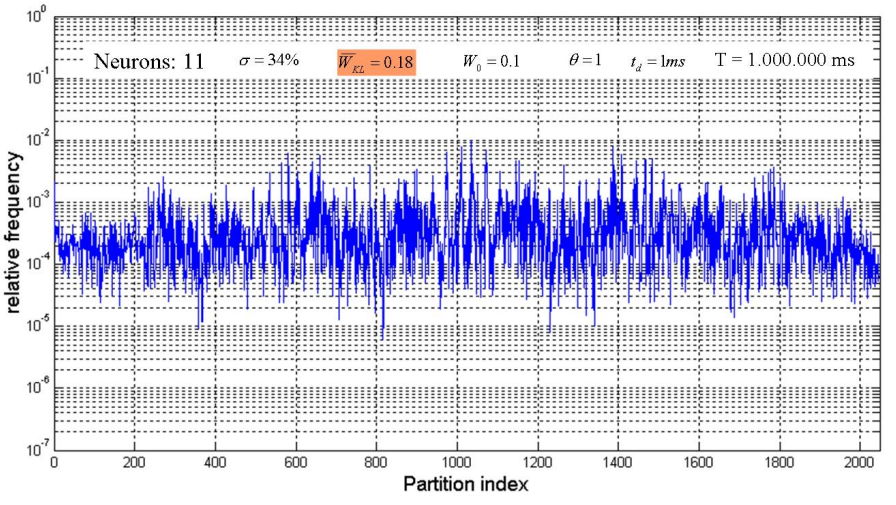


Fig. 2.6 Frequency distribution for $W = 0.18, w = 0.1$

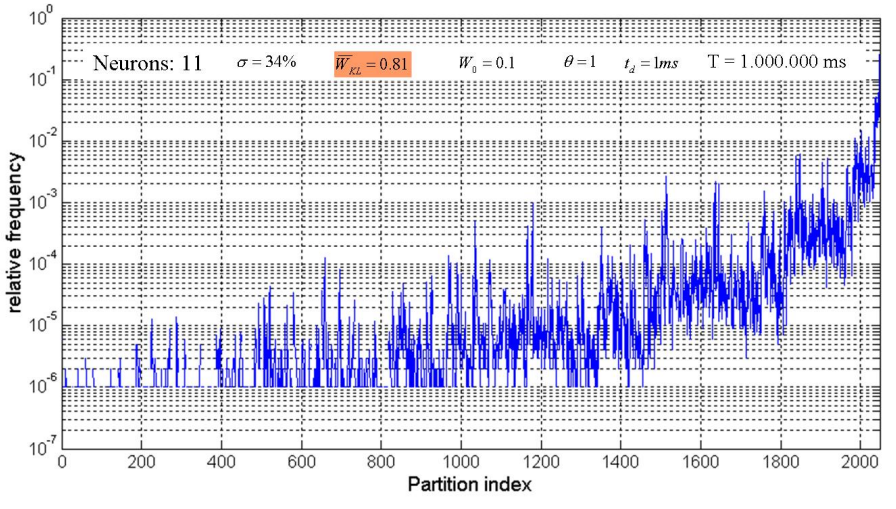


Fig. 2.7 Frequency distribution for $W = 0.81, w = 0.1$

Figure 2.7 shows the phenomenon: to the left of the x-axis the distribution function vanishes completely for some of the partitions. Obviously, the average firing rate must be quite high, and it stands to reason that the frequency distribution may feature little dependence on the values of the input signals.

In Figure 2.8 the average values of the synapses on the input side have increased from 0.1 to 0.5. Figure 2.7 and 2.8 show little differences, their

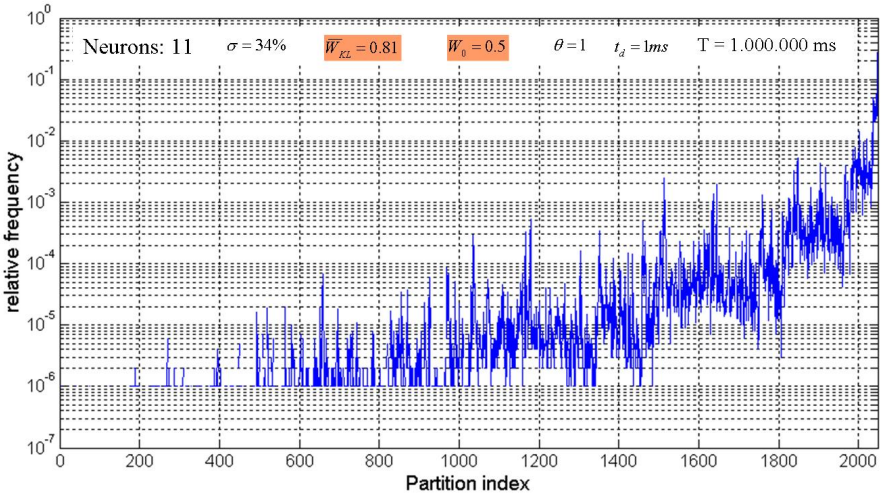


Fig. 2.8 Frequency distribution for $W = 0.81, w = 0.5$

distributions of the frequency of occurrence are practically identical. It is apparent that such „hot" nets are not useful for information processing.

When a net has become too „hot", this can easily be detected. To that end, we investigate the course of the time distances between pulses of any two neurons for a sufficiently long time span T. The histogram of the ISI between two neurons is displayed in Figure 2.9 and 2.10 for three representative (of the 55 possible) pairs of neurons. Figures differ by the values of W, but $w = 0.1$ is always fixed.

The approximate equipartition for $W = 0.018$ is disappearing with increasing values for W. Obviously, the space of all partitions the net lives on

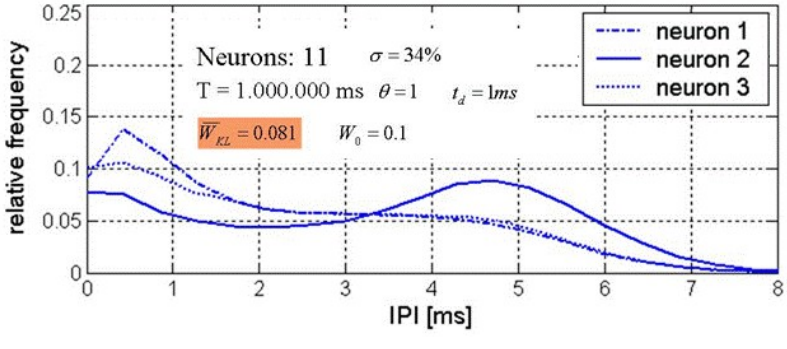


Fig. 2.9 ISI between 2 neurons. 3 representative pairs; $W = 0.081, w = 0.1$.

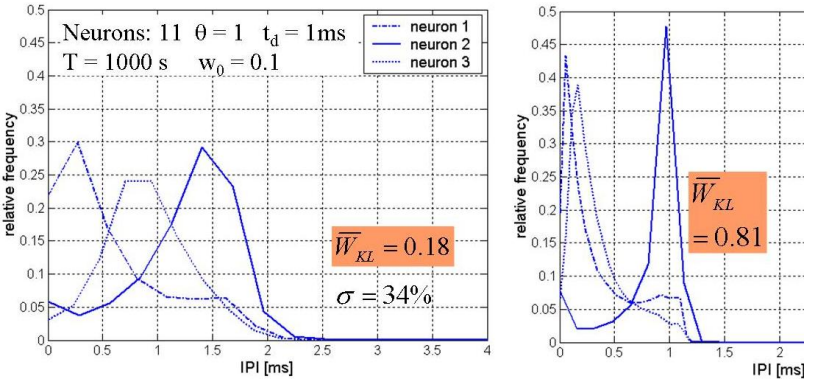


Fig. 2.10 dito, pronounced pattern creation visible

has shrunk. Hence, an observable deviation from equipartition exists which we designate as pattern creation.

As „cold“ net we define a net in which the maximum value of the histogram of all ISI pairs lies within the interval $[0, 0.05]$. If the maximum value exceeds (the values of) the interval, the average value W has to be decreased until the maximum value is within the interval. To be noted is the interval's dependence from the size of the net.

When the experiments which led us from Figure 2.2 till 2.10 are repeated for different values of synapses and initial state variables, the behaviour described above is obtained again, provided mean values of W and w have been chosen sufficiently low:

(1) For constant input and synaptic connections, the net does not respond by a pulse pattern of a single partition or the periodic repetition of the pulses attached to a finite subset of partitions. Rather, the net creates pulses belonging to arbitrary partitions which occur by different frequencies only.
(2) In case of a dependency on the frequency distribution of partitions from the time span T of recording, the time span can always be increased such that the distribution becomes independent from T.
(3) Recorded distributions are slightly dependent on the set of initial values of the net.
(4) The signal dynamics under consideration is not reversible in time: the number and name of pulses accumulating on a neuron's membrane can not be derived from the pulses transmitted by the neuron.

These observations give rise to the reasoning that the statistics of the signals produced by the net delivers the sole set of reproduceably measurable values.

2.4 Entropy and Transfer Function of a Net

One of the central questions of this book is the one of how a net transfers information, and what constitutes information? Thereunder we do not conceive that function which assigns the input signals to the signal values of all neurons at each point in time. For, if we assume constant inputs, many partitions of the net would be traversed, and the constant input signal would be mapped onto a set of pulse vectors varying over time. Furthermore, observation shows that the net will react time-delayed with the creation of a new partition when the input had changed. Hence, a net's transfer function can not be a time function which maps input signals onto output signals at each point in time. Instead, the action of the input signals onto the neurons have to be followed for a while, and mean values have to be found for the characterisation of the net's dynamics.

Let N be the number of neurons in a net. As the above experiments have shown, a set of input signals kept constant for a time span T can be assigned a stationary frequency distribution of partitions. Since for any partition, the logarithm of the frequency distribution plus $log_2 N$ represents the news about this partition, so for knowledge about the diversity of occurrences of the partitions the expectation value of the news is introduced, ie. the news weighted by the frequency distribution. The maximal knowledge would be achieved when the solution of the signal equations (2.1a–c) is known, taking on the value $log_2 N$. Customarily, the maximal knowledge, reduced by the knowledge at the moment, is designated as „non-knowledge" or entropy. Hence, small entropy means a large amount of knowledge, and vice versa.

Thus, the constant input signals can be mapped onto the entropy of the net. To be sure this is the right definition, an experiment has to be devised in which constant inputs are replaced by a set of sequences of pulses. For, we have to connect nets in order to arrive at intelligent functions. Therefore we check whether a stationary frequency distribution produced by a cold net in the time span T and the pulses of which are input to the net under investigation makes the net respond with a stationary frequency distribution. This is the case, indeed, as is shown in Figure 2.11.

Let p_i and p_a, respectively, indicate the stationary frequencies of occurrence for the partitions at the input and output side, respectively. Then,

$$S_{in} = -\sum_i p_i \cdot \ln p_i \qquad \text{and} \qquad S_{out} = -\sum_a p_a \cdot \ln p_a \tag{2.3}$$

represent the entropy at the input and output side, respectively. A reasonable definition of information processing as input entropy to be mapped on to output entropy demands that it is reproduceable.

For that purpose, we choose on the one hand 200 diverse sets of synapses of equal mean value and mean variation, with fixed initial values for the net (Figure 2.12 top) , and on the other hand 200 different sets of initial values $a(t_0) < \theta$ and $\Delta(t_0) < 1\,ms$, with fixed synapses (Figure 2.12 bottom). The

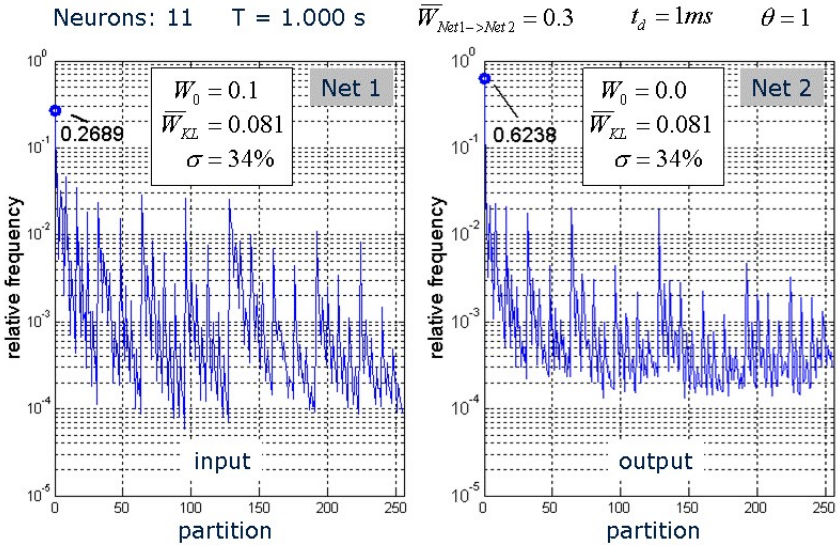


Fig. 2.11 Stationary input distribution creates a similar one at the output side

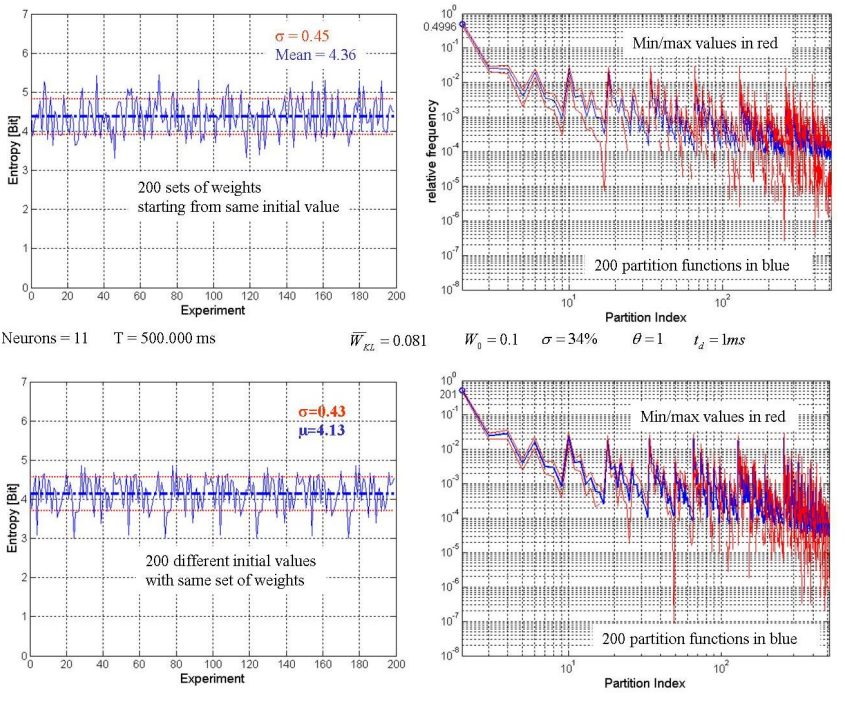


Fig. 2.12 Entropy and frequency distribution of a net with synapses varied (top) and with initial values varied (bottom)

Figures 2.12 show the entropy on the left and the frequency distributions inclusive of maximum and minimum values on the right. Obviously, the entropy of a net so far defined varies a bit, and that is why we want to define entropy as the one defined for an ensemble of nets, ie. averaged over all initial values and all representative sets of synapses with equal mean value and mean variation. In following experiments, entropy and frequency distribution are assumed to be measured or defined in this manner.

In summary, as information-carrying transfer function of a net we can define that function which maps an ensemble-based entropy and frequency distribution, respectively, at the input side in a similar one at the output side.

2.5 Operating Range of a Net

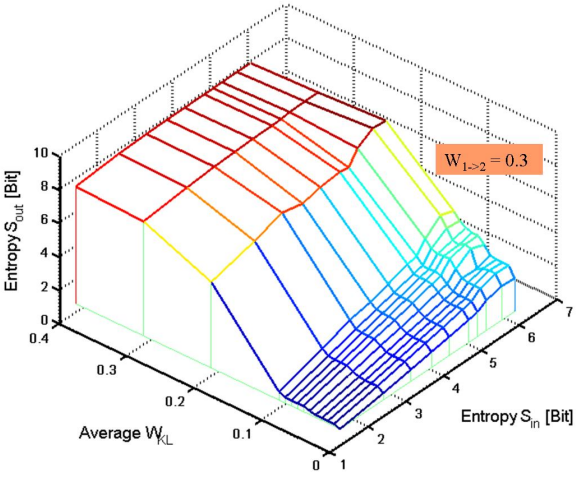
The fact that the entropy is dependent on synapses and input signals is important for all applications. We regard the entropy at the input side as variable, and we are asking for the operating range of synapses such that the entropy remains a useable function of the input signals. For generating S_{in} we use a cold net. In Figure 2.13a the entropy S_{out} is recorded as function of the entropy S_{in} at the input side and of the mean value W of the synapses connecting the neurons. Here all input synapses have the value 0.3. The mean variation of the synapses amounts to 34 percent, the change of initial values and synaptic representatives is carried out 200 times, and the simulation time is 2000 s at a time.

Figure 2.13b differs from Figure 2.13a by all input synapses being set to 0.8. As expected, the entropy saturates for larger values of W. The useable cold operating range results in rather lower values for W.

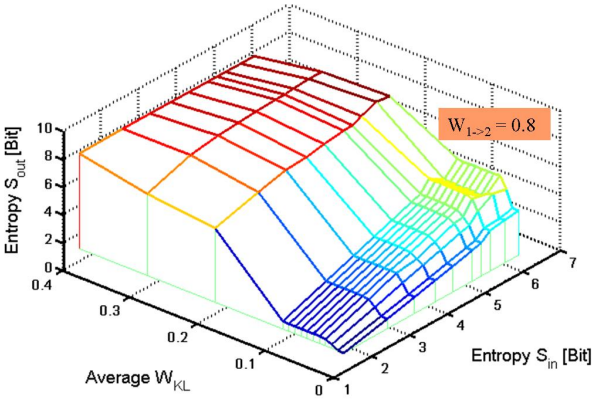
2.6 Pulse Rates

From the frequency distribution of the partitions we can derive various interesting values, among them the pulse rate. On the one hand, the pulse rate of a neuron can be introduced as the average number of pulses per time interval. In order for it to become stationary the measurement period must be sufficiently large. On the other hand, the frequency distribution can be taken to compute the frequency of a neuron n transmitting a pulse. The latter results in the sum P_n of the frequencies of all partitions in which the neuron n participates as transmitter. Then the number of pulses fired by the neuron n in the time span T amounts to $P_n \cdot T/t_d$. Hence, the pulse rate F_n of a neuron n reads:

$$F_n = \frac{P_n}{t_d} \ . \tag{2.4}$$



(a) external coupling $w_{e0} = 0.3$



(b) external coupling increased to $w_{e0} = 0.8$

Fig. 2.13 Entropy as function of input entropy and internal coupling

Just as the entropy the pulse rate is identified over all initial values $a(t_0)$, $\Delta(t_0)$ and representatives of a given synaptic equipartition, ie. in terms of the ensemble treatment[4].

Figure 2.14 (left) shows the pulse rates of the 14 neurons of a fully connected net, with equal input $i = 0.5$ for all neurons. When the number of neurons is increased, the mean variation of the mean pulse rate F (of all neu-

[4] It must be noted that „pulse rate of a net" refers to a single net. When „pulse rate" is used without further reference, that signifies the pulse rate of the ensemble.

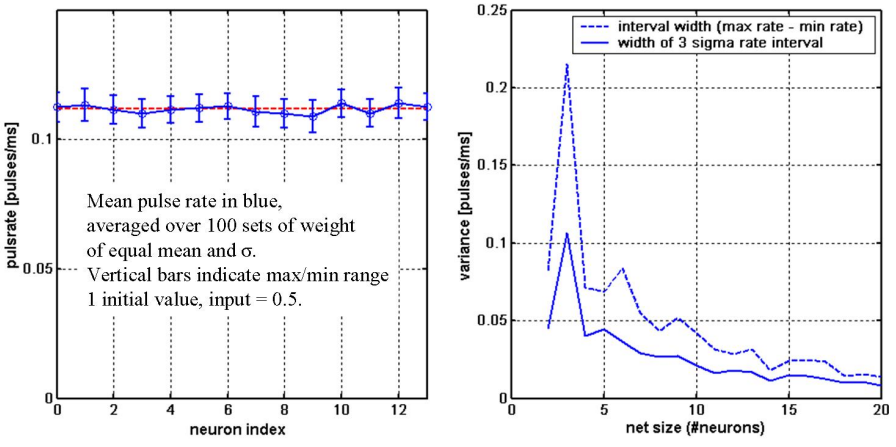


Fig. 2.14 Pulse rates of 14 neurons of a fully connected net (left) and mean variation of the mean pulse rate as function of net size (right)

rons) decreases. This is in agreement with the fact that the resolution of a net increases with the number of neurons in a net.

In the same way defect-tolerance is increased with increasing net size. Figure 2.15 displays the change of the mean pulse rate and its mean variation, when randomly picked neurons (1, 2, 3 or 4 of the 14 neurons) are „dead".

Compared with the net immanent resolution (cf. Figure 2.14) and the limited resolution at the input side, a „graceful degradation" arises.

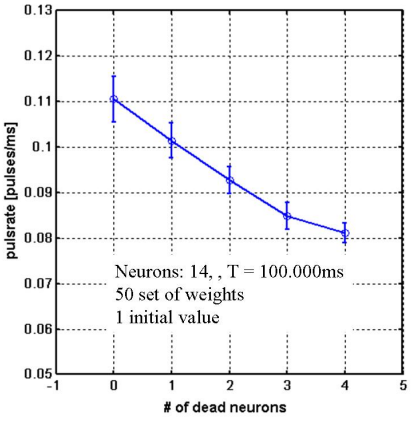


Fig. 2.15 Mean pulse rate and mean variation as function of dead neurons

With the derivation of the entropy as basic function of a net with constant synapses the often used notion of pulse rate is solidly rolled out. Because the entropy is the original variable and the current use of the notion of „pulse rate coding“ lacks the information-theoretic background, the entropy comes to the fore further on, albeit the pulse rate being the variable of practical value for the experimenter and user.

Without offering more data we inform the reader that the results obtained for constant synapses also hold for synapses featuring a fixed function of time or for neurons firing (sequences of) exponentially rising and decaying pulses.

2.7 Resolution and Net Size

Let us consider a fully connected net of 9 neurons which are coupled via synapses (all equal and of constant value 0.3) to a single input signal ranging between 0 and 1. We determine how large the net must grow before a given resolution at the input side is dominated by the net immanent resolution (for instance, represented by the mean variation of the mean pulse rate). Figure 2.16 displays the maximal and minimal mean pulse rate of the net, respectively, as function of the input signal. Both pulse rates are approximated by a straight line. Here, initial values and synapses were varied 10 times each.

As net immanent resolution the relative distance of the gradients of the straight lines is introduced:

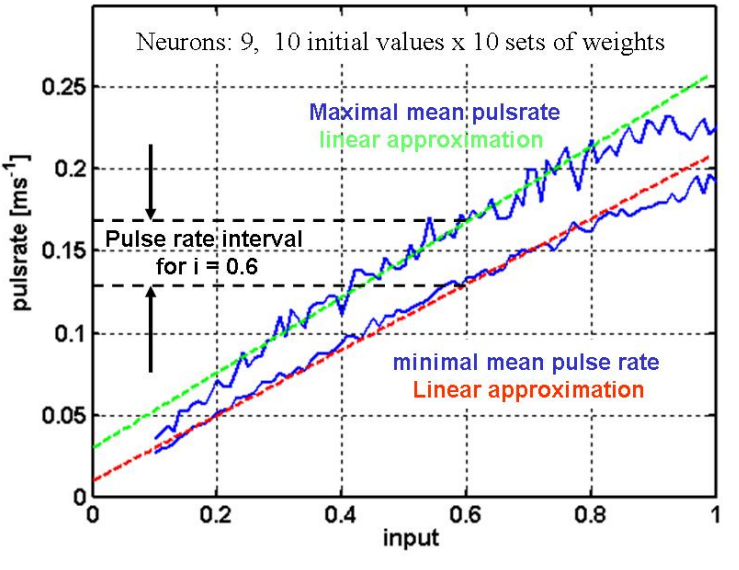


Fig. 2.16 Resolution of the net under consideration

$$A = \frac{green(i) - red(i)}{red(i)} \ .$$

Since a net becomes useless when the resolution at the input side is smaller than the net immanent one, 3 input intervals can be read off from Figure 2.16 which can be separated from one another - irrespective of which initial values or synaptic value sets are chosen: $[0, 0.24]$, $[0.4, 0.6]$ and $[0.8, 1]$. Figure 2.17 shows that the resolution of a net becomes better the more neurons exist in a net.

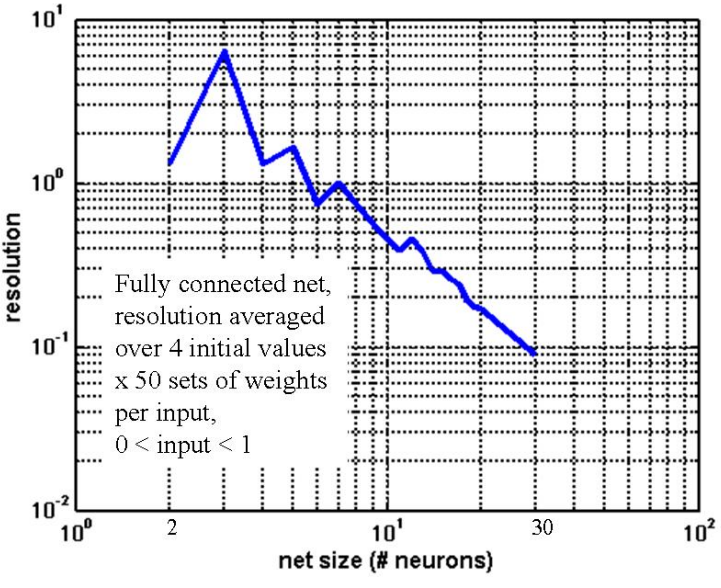


Fig. 2.17 Net size versus resolution

Instead of such a rigorous pointwise separation, a histogram based definition of resolution is recommended as an alternative. For illustration let us consider an input image of 6×6 pixels. The brightness decreases linearly from left to right and from 1 to 0, and in each of the six columns the pixels feature equal brightness. Each pixel inputs to a single neuron, and all neurons are fully connected. The net operates in the cold mode. As a second image we take the one all the pixels of which adopt the mean brightness of the first image. Figure 2.18 (left) shows that a single net (ie. with a fixed synaptic set of weights) can separate the entropies of the images 1 and 2 provided ideal, noise-free images were input and only initial values varied. However, when we input 200 images of type 1 and 2 with 8 percent noise each, already for a single initial value and 4 different sets of representative synapses histograms

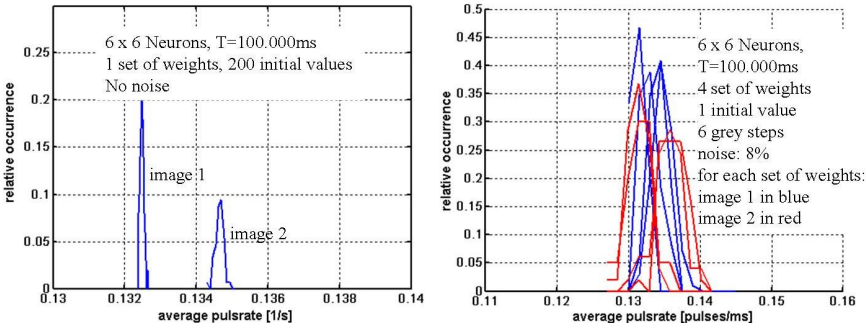


Fig. 2.18 Histogram of pulse rates for images 1,2 without (left) and with noise (right)

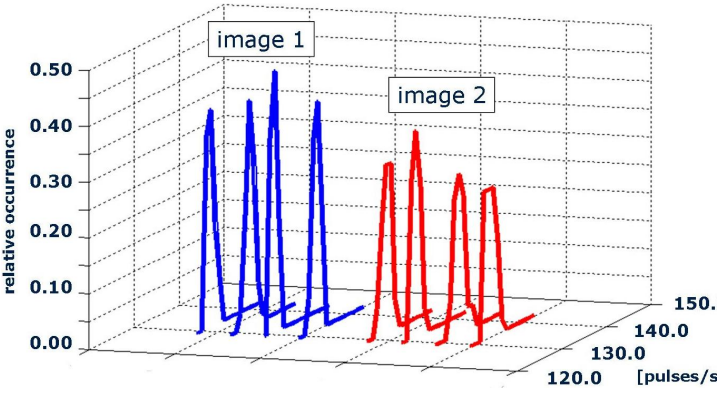


Fig. 2.19 Details of the overlapping in Figure 2.18

do overlap (Figure 2.18 right). More details and insights are presented in Figure 2.19.

Furthermore, a multitude of simulations showed that image 1 and any image created by permutation of the pixels of image 1 overlap, so they are not separable. This is also true for nets which are not fully connected.

In case spacial resolution is required, the size of the net must be increased considerably, in order that the variation of the mean pulse rate becomes sufficiently small, and/or the variation of the synaptic values has to be greatly restricted. The latter could be achieved either by means of so-called learning algorithms, which we do not use for the reasons explained in chapter 1, or by time varying synapses. In chapters 5 and 6 we demonstrate that spacial patterns can be easily and neatly separated with dynamic synapses without reference to any learning algorithms at all.

Let us next consider the time resolution. We ask first for the time it takes the net to produce a stationary pulse rate when starting from a single set of

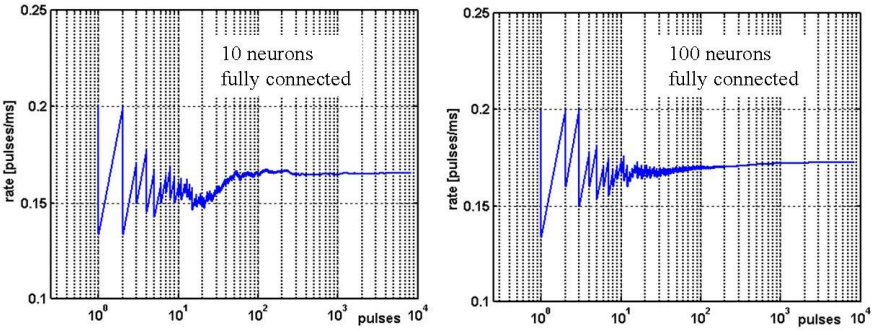


Fig. 2.20 Settling time of a net with 10 (left) and 100 (right) neurons

initial values. It depends a little on the net's size (Figure 2.20) and amounts to about 600 ms.
Secondly, we ask about the time of reaction when the net of 30 fully connected neurons is in stationary mode and the input signal made to jump from 1 to 0.8. It depends on the integration time which is used for the measurement. With about 180 ms a clean and fast separation is obtained (Figure 2.21), ie. the variations of the mean pulse rates for the two signal values do not overlap. In case we wish to increase the signal resolution - jump from 1 to 0.9 - the growth of the net does not help. Instead the integration time must be enlarged.

Therefore, let us summarise:

1. Constant synapses are hardly suitable for the separation of spacial patterns, for number and size of the nets under consideration would grow very rapidly with the necessary degree of spacial resolution.
2. The smaller the time resolution, the less signal resolution is left available.

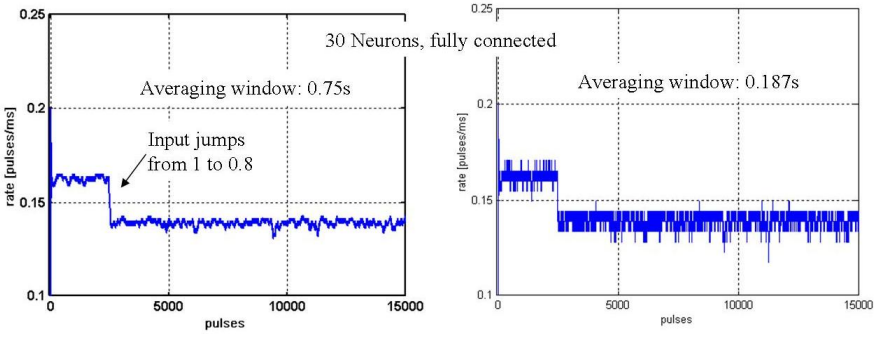


Fig. 2.21 Time of reaction of a net. Pulse rates are measured with different time windows for integration.

2.8 Application Potential

The concatenation of nets, by which the outputs of one net provide the input for the subsequent one, do not prove to be particularly useful. Nevertheless it is instructive. For a great many nets, the last one will produce an entropy value which is close to or beyond the „noise“, ie. in the lower part of the S curve (cf. Figure 2.22, cf. also Figure 2.13). In this manner there is a natural screening between nets which are not directly connected.

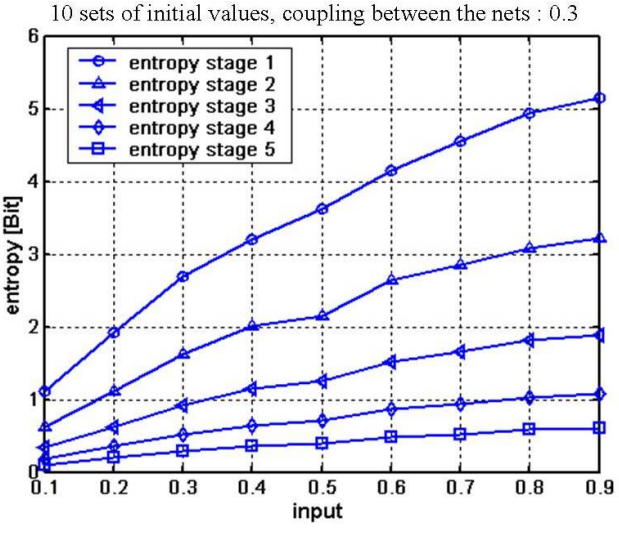
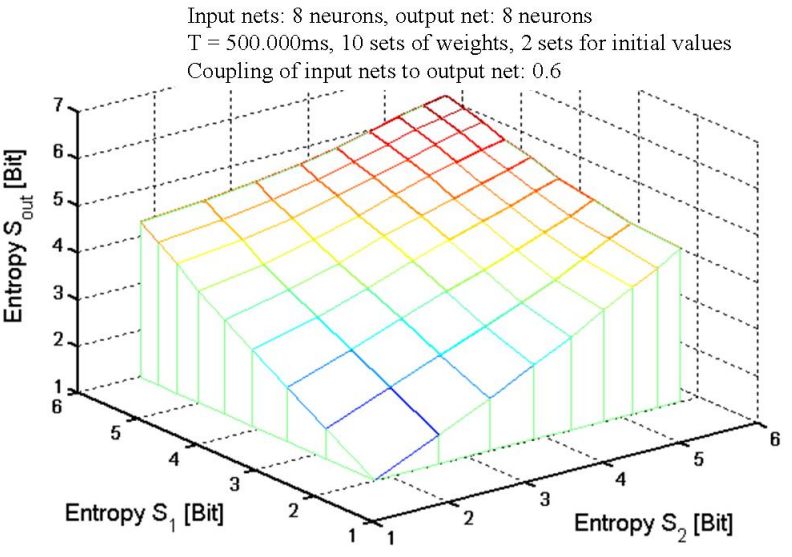


Fig. 2.22 Reduction of entropy within a cascade of equal nets

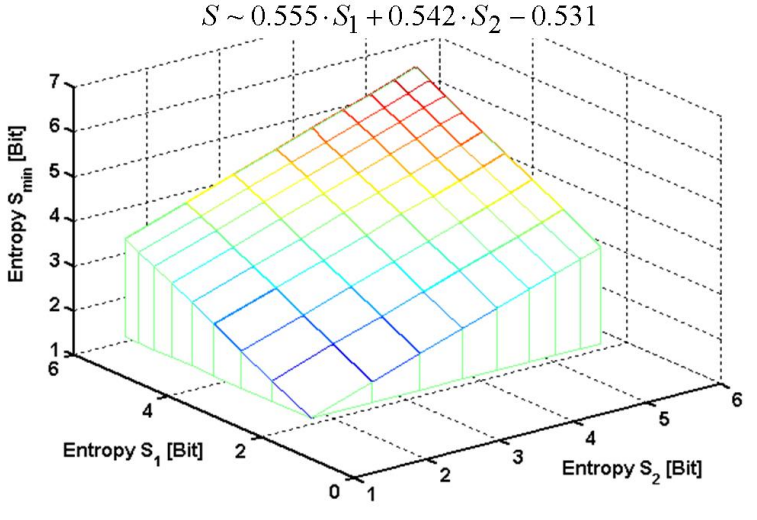
An useful basic arrangement is provided by connecting the outputs of nets N_1 and N_2 with the input synapses of a single fully connected net N. Let us designate the entropies of net N_1 and N_2 by S_1 and S_2. In Figure 2.23a the entropy of the net N is recorded as function of S_1 and S_2 (nets N_1 and N_2 operate in the cold range).

Since the plane $S = 0.551\,S_1 + 0.542\,S_2 - 0.531$ represents a sufficient approximation (Figure 2.23b) - remembering the inherent resolution potential at the input and output side - the net can be characterised as a linear mixer of entropies. We will see in chapter 5 that such nets are an important ingredient of feature detectors.

Let us next turn to a fully connected net which is fostered by a single input (all input synapses of equal value). The input signal ranges between 0 and 1, and the net's resolution is determined by the number of neurons. In the useable cold operating range the mean pulse rate is shown to be an almost



(a)



(b)

Fig. 2.23 (a) Mixing of entropies, (b) linear approximation with 13% error

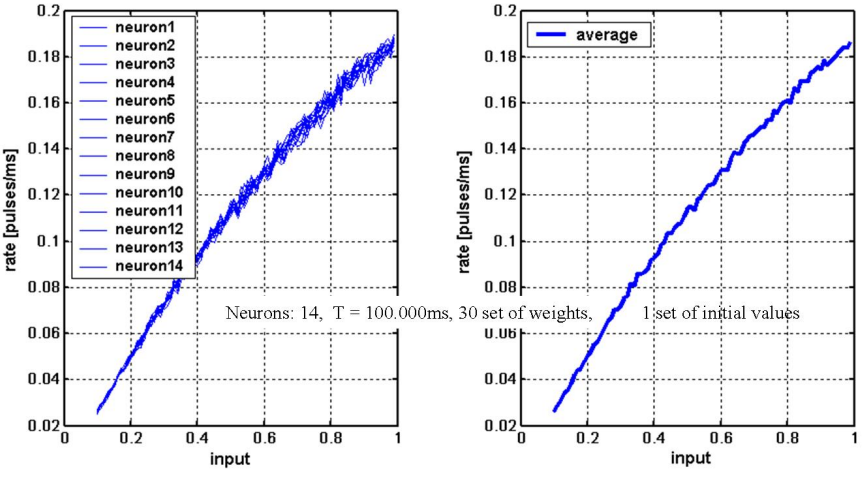


Fig. 2.24 Representative pulse rates (left), mean pulse rate as function of the input (right)

linear function of the input (Figure 2.24). Because the mean variation is also very low, every neuron of the net can be contacted to produce a pulse rate which differs from the mean pulse rate by a very little amount. Contemplating the problem of building an exact synaptic connection between two nets, this is quite a helpful feature of larger nets.

Next, this net will be used for controlling an idealised muscle. In order to copy the action of a muscle - contraction of its length and so doing some work - we use a multitude of springs arranged in sequence. Each such sequence of springs we call a muscle fibre, many of which are working in parallel to form the „muscle“ (Figure 2.25).

The dynamics of a spring is given by:

$$\frac{dx_i}{dt} = -\gamma \cdot (x_i(t) - q_i(t)) - \beta \cdot (2x_i - x_{i+1} - x_{i-1}) \ . \tag{2.5}$$

Each of the 20 muscle fibers is comprised of 5 springs; parallel fibers are coupled stiffly together, hence the contraction of a fiber is damped by the surrounding springs. The 100 springs are each connected to a neuron, and we assume that a spring contracts proportionately to the pulse rate of the attached neuron. Also, the net is operating in the cold range. Figure 2.26 shows that the mean pulse rate of the net strictly follows the input signal (left) and the „muscle“ contracts and relaxes in accordance with the input signal. The same result is obtained if pulses are used for the constant input of the net. Then, the contraction is proportionate to the input pulse rate.

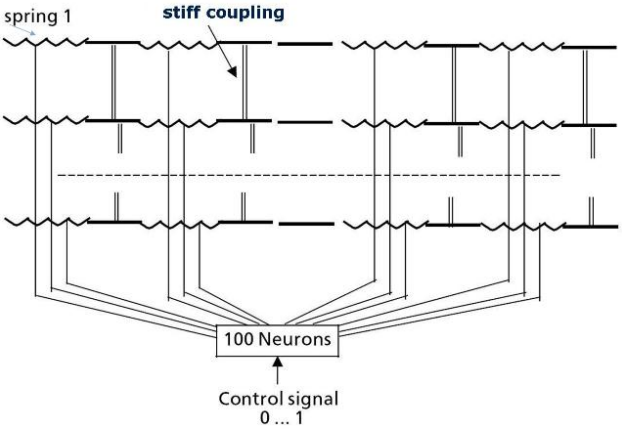


Fig. 2.25 Abstract muscle model

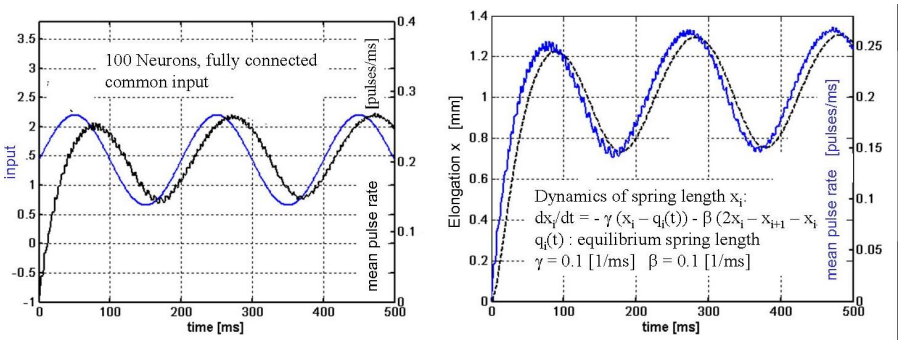


Fig. 2.26 Pulse rate follows input (left), muscle's contraction follows pulse rate (right)

Motor control appears to be another important application of nets with constant synapses. This is even more valid since these nets do not seem able to perform pattern recognition in an economic fashion. As entropy is the only information-carrying function of a net with constant synapses, input entities like images can only be differentiated and separated, respectively, by the values of the entropy. However, the net immanent resolution is sufficiently high and does not interfere with the separation process if the net is extremely huge and the values of the synapses vary only very little. The latter could be achieved by learning algorithms which we consider to be artificial, however. As will be shown in chapter 5 - chapter 8, dynamic synapses introduce a new quantity, the correlation, and thereby provide a much simpler access to pattern recognition.

2.9 Limited Simulation Time

In order that the histograms for the partitions become stationary, the frequency distribution must be determined up to $1 \cdot 10^{-6}$ (cf. Figure 2.5). This requires at least 200000 partitions. Remembering the fact that frequency distributions must be computed not only for a net with fixed initial values and synapses, but for the ensemble, the simulation would easily take a day. For larger nets with > 30 neurons the main memory of a PC does not provide sufficient capacity, so a simulation may become impossible.

When measuring pulse rates instead of entropies, at least 1000 pulse events per neuron are necessary in order to achieve a stable pulse rate (Figure 2.20). Consequently, the memory space needed for the storage of synaptic values of a net limits the simulation. Today, about 1000 fully connected neurons can be simulated on a PC.

Since for larger nets a simulation is out of the question, this gives rise to the desire to treat these nets theoretically. The next chapter starts the theoretical modelling. It covers nets with constant as well as dynamic synapses.

2.10 Summary

Since brains can receive information about an image, the first important question is about the information-carrying function of a net of pulse neurons. A first answer is obtained for nets with constant synapses, or synapses the behaviour of which is controlled by a fixed time function. In this case the occurrence in time of pulses produced by the neurons is not reproduceable, but only the frequency of occurrence. Therefore, we are forced to combine exactly defined knowledge formed by the signal equations with statistical quantities, namely the expectation values for inter-spike intervals and number of pulses.

Nets with constant synapses are characterised by mean values: pulse rate incl. its mean variation for each neuron, or mean pulse rate of the net and the ensemble, respectively, incl. its mean variation. All the statistics are based upon the frequency with which the partitions of a net of N neurons into ε receiving neurons and $N-\varepsilon$ transmitting ones will occur, $0 \leq \varepsilon \leq N$. In order to obtain reproduceable quantities, an ensemble of nets must be considered and the frequencies recorded over all sets of initial values and all sets of synaptic values having equal mean value and variation. With the resulting frequency distribution of the partitions entropy, pulse rate, and time and spacial signal resolution can be derived. Therefore, the information-carrying function of a net is the entropy, which itself is a function of the entropy resulting from the pulses at the input inside.

Introducing a quantity like the entropy as a basic function accords with the assumption made from the earliest days of brain research, that neural nets possess a processing of signals which is defect-tolerant. Indeed, this property, often cited as robustness, could be proven (Figure 2.15).

In order to use nets with constant synapses, they have to be operated in the „cold“ range. With properly chosen values for threshold and pulse duration, pulse rates from a few Hertz up to several hundred Hertz can be achieved. In this way it is guaranteed that the pulse activity of any pair of neurons in the net is not correlated and, hence, no pattern can be formed. Note that an artificial pulse pattern can always be created with sufficiently high input signals or synaptic coupling between neurons.

Furthermore, the size of a net is an important design criterion, for the precision of pulse processing achieved by a net is determined by the resolution available for entropy and pulse rate, respectively. The resolution of both quantities is a function of the signal resolution on the input side, of the chosen set of synapses, and the number of neurons. The larger the resolution of the input signals is set, the larger the nets and the smaller the variation of synaptic values which have to be chosen.

In consequence, on the one hand nets with constant synapses are extremely limited in their capacity to resolve time and spacial patterns. In particular, two-dimensional input images can often be re-combined pixel-wise, although entropy and pulse rate do not change significantly. On the other hand, nets with constant synapses excel in controlling of muscles and alike.

Since patterns transport valuable information, there must exist another type of pulsing nets besides the ones we have considered so far.

3

Theory of Nets with Constant or Dynamic Synapses

Ulrich Ramacher

The experimental observation of the presence and duration of partitions and pulse patterns, respectively, is one alternative to achieve the characterisation of a stationary frequency of distribution. For larger networks, however, the simulation time exceeds our human patience, hence a theoretic coverage is necessary. As the constant synapses form a special case, the development of a generic theory of nets with dynamic synapses becomes necessary. To this end we will introduce the idea of the *signal energy* and show that the signal equations as well as the entropy are derivable from it. This way the state trajectories of a net can be obtained by theory, for example as the dependency of the entropy on the input pulses. The theory developed here holds for arbitrary input signals (> 0), but pulses of amplitude x and duration t_d only. The reason for the latter restriction is the much more complicated calculations that would be required, and for the purposes of this book, it would not add anything regarding pulse signals in the treatment of image recognition tasks.

Having developed the generic theory we will focus on the case of constant synapses in chapter 4 and examine the theory first with respect to the experiments presented in chapter 2.

3.1 Derivation of the Signal Energy

It suffices to define the signal energy for every partition, because at any time the net is active in one of the 2^N possible partitions into receiving and/or transmitting neurons. Let us consider an arbitrary partition featuring receiving neurons. Then, a total of $\binom{N}{\varepsilon}$ such partitions with exactly ε receiving neurons exists which can be enumerated by $i = 1 \ldots \binom{N}{\varepsilon}$. Such a particular partition will be designated by $\{\varepsilon i\}$, its duration by $\tau_{\varepsilon i}$, and the set of receiving neurons and transmitting ones, respectively, living at time $t \in [t_{\varepsilon i}, t_{\varepsilon i} + \tau_{\varepsilon i})$ by $E_{\varepsilon i}$ and $S_{\varepsilon i}$, resp. All quantities are chosen dimensionless.

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 47–58, 2010.
© Springer-Verlag Berlin Heidelberg 2010

The signal energy is deduced for the generic case (2.1c) of dynamic synapses. In order to simplify the calculations, the term Q_{es} is used in a specialized sense which is sufficient for the applications considered in this book[1]:

$$\begin{aligned} &Q_{es}(t) = \mu \cdot (a_e(t) - \theta/2) \qquad \text{for } e \in E_{\varepsilon i}, s \in S_{\varepsilon i} \cap N_e, \\ &Q_{es}(t) = 0 \qquad \text{otherwise .} \end{aligned} \tag{3.1}$$

The signal energy is thought to be composed of, firstly, a synapse energy and, secondly, a membrane energy. As synapse energy we define:

$$\frac{1}{2} \sum_{e \in E_{\varepsilon i}} \left\{ \sum_{s \in S_{\varepsilon i} \cap N_e} W_{es}(t)^2 \cdot x_s + W_{e0}(t)^2 \cdot i_e(t) \right\} + \frac{1}{2} \sum_{e \in E_{\varepsilon i}} \sum_{s \in E_{\varepsilon i} \cap N_e} W_{es}(t)^2 \ , \tag{3.2a}$$

with the second term representing the rest energy of the synapses. As total energy of the pulses stored at the membrane we define:

$$\frac{1}{2} \sum_{e \in E_{\varepsilon i}} (a_e(t) - \theta/2)^2 \ . \tag{3.2b}$$

Obviously, the membrane signals are composed of pulses belonging to earlier partitions. Hence we obtain for the membrane signal of the e-th receiving neuron:

$$a_e(t) = \sum_{z=1}^{z_e} \sum_{s \in \{S_{\varepsilon i(z)} \cap N_e\} \cup \{0\}} M_{es}(t_{\varepsilon i(z+1)}) \ .$$

Here M_{es} is determined by

$$\begin{aligned} &M_{es}(t_{\varepsilon i(z+1)}) := \int_{t_{\varepsilon i(z)}}^{t_{\varepsilon i(z+1)}} W_{es}(t') \cdot x_s dt' \qquad \text{or} \\ &M_{e0}(t_{\varepsilon i(z+1)}) := \int_{t_{\varepsilon i(z)}}^{t_{\varepsilon i(z+1)}} W_{e0}(t') \cdot i_e(t) dt' \ , \\ &t_{\varepsilon i(z_e+1)} = t, \quad t_{\varepsilon i(z_e)} = t_{\varepsilon i}, \ldots, t_{\varepsilon i(1)} = t_e \ , \end{aligned} \tag{3.2c}$$

with z counting the number z_e of the partitions $\{\varepsilon i(z)\}$ which occurred between the (re-)start of the accumulation process of the e-th neuron and the

[1] Although the structural similarity of this expression to the well-known *Hebb's rule* is apparent, the challenge consists in the definition of its exact form. As will be seen in chapter 5 it is the possibility of the growing and decaying of the synaptic values - implied by the reference point $a_e = \theta/2$ and the right choice of the sign of μ - which enables intelligent information processing. This rule was found as early as April 1999 by U. Ramacher and has proved to be extremely valuable [41, 42]. It enables such different things as image segmentation, binding of features to objects or calculation of pulse differences.

current partition $\{\varepsilon i\}$. Besides the above (e,s)-combinations we also consider the following ones:

$$e \in E_{\varepsilon i}, s \in E_{\varepsilon i} \cap N_e \ : M_{es}(t) := \int_{t_{\varepsilon i(z)}}^{t} W_{es}(t')\, dt' \ , \tag{3.2d}$$

however, we do not introduce an additional potential energy term for these variables.

As signal energy $h_{\varepsilon i}$ of a partition we can now define:

$$h_{\varepsilon i}(t) := \frac{1}{2} \sum_{e \in E_{\varepsilon i}} \sum_{s \in \{S_{\varepsilon i} \cap N_e\}} W_{es}(t)^2 \cdot x_s + \frac{1}{2} \sum_{e \in E_{\varepsilon i}} W_{e0}(t)^2 \cdot i_e(t) + \frac{1}{2} \sum_{e \in E_{\varepsilon i}} \sum_{s \in E_{\varepsilon i} \cap N_e} W_{es}(t)^2 - \frac{\mu}{2} \sum_{e \in E_{\varepsilon i}} (a_e(t) - \theta/2)^2 \ . \tag{3.3}$$

It should be noted that the signal energy is zero for the partition with transmitters only.

Considering the derivative of the signal energy $h_{\varepsilon i}$ with respect to the independent variables W_{es} and M_{es} for all (e,s)-combinations we obtain:

$$\begin{aligned}
e \in E_{\varepsilon i}, s \in \{S_{\varepsilon i} \cap N_e\} : \ & \frac{\partial h_{\varepsilon i}}{\partial M_{es}} = -\mu \cdot (a_e(t) - \theta/2) \ , \ \frac{\partial h_{\varepsilon i}}{\partial W_{es}} = W_{es}(t) \cdot x_s \\
e \in E_{\varepsilon i} : \ & \frac{\partial h_{\varepsilon i}}{\partial M_{e0}} = -\mu \cdot (a_e(t) - \theta/2) \ , \ \frac{\partial h_{\varepsilon i}}{\partial W_{e0}} = W_{e0}(t) \cdot i_e(t) \\
e \in E_{\varepsilon i}, s \in E_{\varepsilon i} \cap N_e : \ & \frac{\partial h_{\varepsilon i}}{\partial M_{es}} = 0 \, , \qquad \frac{\partial h_{\varepsilon i}}{\partial W_{es}} = W_{es}(t) \ .
\end{aligned}$$

Clearly, the relation holds:

$$e \in E_{\varepsilon i}, s \in \{S_{\varepsilon i} \cap N_e\} \cup \{0\} \cup \{E_{\varepsilon i} \cap N_e\} : \frac{\partial h_{\varepsilon i}}{\partial W_{es}} = \frac{dM_{es}}{dt} \ , \tag{3.4a}$$

meaning that a change of W causes a change of M by means of h. Conversely, in the sense of actio and re-actio, a change of M should influence W by means of h. Hence, we make the ansatz:

$$e \in E_{\varepsilon i}, s \in \{S_{\varepsilon i} \cap N_e\} \cup \{0\} \cup \{E_{\varepsilon i} \cap N_e\} : -\frac{\partial h_{\varepsilon i}}{\partial M_{es}} = \frac{dW_{es}}{dt} \ . \tag{3.4b}$$

This way the following equations result:

$$\begin{aligned}
e \in E_{\varepsilon i}, s \in \{S_{\varepsilon i} \cap N_e\} \cup \{0\} : \qquad & \mu \cdot (a_e(t) - \theta/2) = \frac{dW_{es}}{dt} \ , \\
e \in E_{\varepsilon i}, s \in \{E_{\varepsilon i} \cap N_e\} : \qquad & 0 = \frac{dW_{es}}{dt} \ ,
\end{aligned}$$

which agree with the signal equations (2.1c), except for the decay term.

Let us look back for a moment: it seems that the expression for the signal energy in equation (3.3) has been chosen quite arbitrarily, and the equation (3.3) was a matter of guesswork, all for the purpose of deriving the original

signal equations from the signal energy. Therefore we have to assure ourselves that this procedure can be embedded into a mathematical framework. Indeed, the signal equations (3.4a, b) are also obtained when we start with a partial differential equation of first order for a function $J_{\varepsilon i}(t, M_{es})$:

$$\frac{\partial J_{\varepsilon i}(t, M_{es}, M_{e0})}{\partial t} + h_{\varepsilon i}(W_{e0}, M_{e0}, M_{es}, W_{es}) = 0 \ .$$

Here M_{es} is a variable which is not defined by (3.2c) (but shown to fulfil it) and $W_{es} := \partial J_{\varepsilon i}/\partial M_{es}$ the so-called conjugate variable of M_{es} defined for all (e, s)-combinations. The solution of this partial differential equation is given by the solution of the set of characteristic functions [43]:

$$\begin{aligned}
&\frac{dM_{es}}{dt} = \frac{\partial h_{\varepsilon i}}{\partial W_{es}}, \frac{dW_{es}}{dt} = -\frac{\partial h_{\varepsilon i}}{\partial M_{es}}, e \in E_{\varepsilon i}, s \in \{S_{\varepsilon i} \cap N_e\} \cup \{0\} \cup \{E_{\varepsilon i} \cap N_e\} \ , \\
&\frac{d(\partial J_{\varepsilon i}/\partial t)}{dt} = -\frac{\partial h_{\varepsilon i}}{\partial t} \ , \\
&\frac{dJ_{\varepsilon i}}{dt} = \frac{\partial J_{\varepsilon i}}{\partial t} + \sum_{e \in E_{\varepsilon i}} \left[\sum_{s \in \{S_{\varepsilon i} \cap N_e\} \cup \{0\}} W_{es}(t) \frac{dM_{es}}{dt}(t) + \sum_{s \in \{E_{\varepsilon i} \cap N_e\}} W_{es}(t) \frac{dM_{es}}{dt}(t) \right] \ .
\end{aligned}$$

The first line of this set of equations agrees with (3.4a, b). In particular, the M-variables result as defined in (3.2c-d). The remaining characteristic equations help determine the function J, which will not be made use of in this book, however. In summary, we have embedded the signal energy into a partial differential equation the characteristic equations of which deliver the signal equations (2.1c), except for the decay term. The above partial differential equation can, however, be generalized to include a term that creates the decay term in (2.1c).

With the partial differential equation of first order

$$\frac{\partial J_{\varepsilon i}}{\partial t} + \gamma \cdot J_{\varepsilon i}(t, M_{e0}, M_{es}) + h_{\varepsilon i}(W_{e0}, M_{e0}, M_{es}, W_{es}) = 0 \tag{3.5a}$$

the characteristic equations read:

$$\begin{aligned}
&\frac{dM_{es}}{dt} = \frac{\partial h_{\varepsilon i}}{\partial W_{es}} \ , \frac{dW_{es}}{dt} = -\gamma \cdot W_{es}(t) - \frac{\partial h_{\varepsilon i}}{\partial M_{es}} \ , \\
&\qquad\qquad e \in E_{\varepsilon i}, s \in \{S_{\varepsilon i} \cap N_e\} \cup \{0\} \cup \{E_{\varepsilon i} \cap N_e\} \ , \\
&\frac{d(\partial J_{\varepsilon i}/\partial t)}{dt} = -\gamma \cdot \frac{\partial J_{\varepsilon i}}{\partial t} - \frac{\partial h_{\varepsilon i}}{\partial t} \ , \\
&\frac{dJ_{\varepsilon i}}{dt} = \frac{\partial J_{\varepsilon i}}{\partial t} + \sum_{e \in E_{\varepsilon i}} \left[\sum_{s \in \{S_{\varepsilon i} \cap N_e\} \cup \{0\}} W_{es}(t) \frac{dM_{es}}{dt}(t) + \sum_{s \in \{E_{\varepsilon i} \cap N_e\}} W_{es}(t) \frac{dM_{es}}{dt}(t) \right] \ .
\end{aligned} \tag{3.5b}$$

Specifically, the first line of (3.5b) gives the generic form of the synaptic equation:

$$\begin{aligned}\frac{dW_{es}(t)}{dt} &= -\gamma \cdot W_{es} + \mu \cdot (a_e(t) - \theta/2) \ , \quad e \in E_{\varepsilon i}, s \in \{S_{\varepsilon i} \cap N_e\} \cup \{0\} \ , \\ \frac{dW_{es}(t)}{dt} &= -\gamma \cdot W_{es} \ , \qquad e \in E_{\varepsilon i}, s \in \{E_{\varepsilon i} \cap N_e\} \ .\end{aligned} \tag{3.6}$$

These equations agree with (2.1c). The other signal equations (2.1a, b) are contained already in the definition (3.3) of $h_{\varepsilon i}$: in (3.2b) the membrane potential of a neuron is used in accordance with (2.1a), furthermore the pulse signal $x_s(t)$ which fulfils (2.1b) is part of $h_{\varepsilon i}$, and finally the phases of accumulation and pulse transmission are taken care of by indexing the signal energy by means of the partition $\{\varepsilon i\}$ in the life time of which the signal equations (2.1a-c) hold[2].

Hence, the signal equations (2.1a-c) are uniquely derived from a partial differential equation of the first order. Moreover, the partial differential equation is uniquely defined by the signal energy h. As in chapter 2 the entropy was derived statistically from the signal equations and now the signal equations have been obtained from the signal energy, so we expect to derive the entropy directly from the signal energy. Because the temporal mean of h can be reconstructed by means of the frequency distribution of the variables W, M and the entropy is an expectation value with the frequency distribution as density, we have a formal connection between signal energy and entropy, at least.

3.2 Temporal Mean and Spatial Mean

The mean signal energy H can be computed experimentally by solving the signal equations at any time t - starting with t_0 and finishing at T - and inserting the solution into the expression for the signal energy and thereby determining the temporal mean. However, in order to obtain a structured expression showing the influence of synapses and inputs on the temporal mean of h, we analyse the sum of averaged signal energies of those partitions occurring in the interval $[t_0, T]$. Let the quantity $B_{\varepsilon i}$ designate how often the partition $\{\varepsilon i\}$ occurred during the observation time T. Further, let $t_{\varepsilon i(\nu)}$ denote the start time of the ν-th occurrence and $\tau_{\varepsilon i(\nu)}$ the duration of this partition (remember: with the exception of the partition with only receiving neurons each partition lasts at most as long as a pulse).

Then we receive as temporal mean of the energy:

[2] The equation (3.5a) without the decay term is known in Physics as the Hamilton-Jacobi equation, the term h as Hamiltonian, which constitutes the energy function of the system under consideration (clearly, with different variables). The concept of Hamiltonian is basic for all areas of Physics. In particular, it connects mechanics, thermodynamics, and quantum mechanics.

$$H(t_0,T)=\sum_{\varepsilon=0}^{N}\sum_{i=1}^{\binom{N}{\varepsilon}}\frac{1}{T}\sum_{\nu=1}^{B_{\varepsilon i}}\int_{t_{\varepsilon i(\nu)}}^{t_{\varepsilon i(\nu)}+\tau_{\varepsilon i(\nu)}} h_{\varepsilon i}(M_{\varepsilon i}(t),W_{\varepsilon i}(t),i_{\varepsilon i}(t),t_{\varepsilon i(\nu)})\,dt\ , \tag{3.7}$$

with $W_{\varepsilon i}$, $M_{\varepsilon i}$ indicating synapses and membrane signal portions present in the partition $\varepsilon i(\nu)$. H can be computed by this equation and the signal equations (2.1a-c) in the case of small nets (< 30 neurons). For larger nets the further interpretation of (3.7) takes place by transforming temporal integrals into spatial ones (in other words: histograms) by means of a frequency distribution $\phi_{\varepsilon i(\nu)}(M_{\varepsilon i},W_{\varepsilon i},i_{\varepsilon i};t_{\varepsilon i(\nu)},\tau_{\varepsilon i(\nu)})$ which counts how often synapses and membrane signals occurred in the partition $\{\varepsilon i(\nu)\}$ during $[t_0,\,T]$:

$$\begin{aligned}H(t_0,T)=\sum_{\varepsilon=0}^{N}\sum_{i=1}^{\binom{N}{\varepsilon}}\sum_{\nu=1}^{B_{\varepsilon i}}\int di_{\varepsilon i}\,dM_{\varepsilon i}\,dW_{\varepsilon i}\,\phi_{\varepsilon i(\nu)}(M_{\varepsilon i},W_{\varepsilon i},i_{\varepsilon i};t_{\varepsilon i(\nu)},\tau_{\varepsilon i(\nu)})\\ \cdot\, h_{\varepsilon i(\nu)}(M_{\varepsilon i},W_{\varepsilon i},i_{\varepsilon i};t_{\varepsilon i(\nu)})\ .\end{aligned} \tag{3.8}$$

The following abbreviations are used:

$$\begin{aligned}\int di_{\varepsilon i}\,dM_{\varepsilon i}\,dW_{\varepsilon i}=\prod_{e\in E_{\varepsilon i}}\int di_e\cdot\prod_{e\in E_{\varepsilon i}}\prod_{s\in\{S_{\varepsilon i}\cap N_e\}\cup\{0\}}\int dM_{es}\,dW_{es}\\ \cdot\prod_{e\in E_{\varepsilon i}}\prod_{s\in\{E_{\varepsilon i}\cap N_e\}\cup\{0\}}\int dW_{es}\ .\end{aligned}$$

For the sake of clarity, the integral limits will only be inserted when concrete calculations have to be performed.

It must be noted that the input signals i_e may represent constants, any fixed temporal functions or pulse signals. Furthermore, input signals may be unknown as to their temporal progression or - in case of pulses - as to their temporal occurrence. Hence, they will appear as parameter functions of the frequency distribution, and pulse rates measured at the input side must be included.

In the following, the dependency of $\phi_{\varepsilon i(\nu)}$ on its parameters $t_{\varepsilon i(\nu)}$ and $\tau_{\varepsilon i(\nu)}$ will be suppressed, again for the sake of readability.

It results from

$$1=\sum_{\varepsilon=0}^{N}\sum_{i=1}^{\binom{N}{\varepsilon}}\frac{1}{T}\sum_{\nu=1}^{B_{\varepsilon i}}\int_{t_{\varepsilon i(\nu)}}^{t_{\varepsilon i(\nu)}+\tau_{\varepsilon i(\nu)}} dt \tag{3.9}$$

and by passing to the spatial integral that the probability $P_{\varepsilon i(\nu)}$ of occurrence of the partition $\varepsilon i(\nu)$ is given by:

$$\frac{\tau_{\varepsilon i(\nu)}}{T}=\int di_{\varepsilon i}\,dM_{\varepsilon i}\,dW_{\varepsilon i}\,\phi_{\varepsilon i(\nu)}(M_{\varepsilon i},W_{\varepsilon i},i_{\varepsilon i})=:P_{\varepsilon i(\nu)}\ . \tag{3.10}$$

As signal entropy we use:

$$S_0 := -\sum_{\varepsilon=0}^{N}\sum_{i=1}^{\binom{N}{\varepsilon}}\sum_{\nu=1}^{B_{\varepsilon i}}\int di_{\varepsilon i}\, dM_{\varepsilon i}\, dW_{\varepsilon i}\, \phi_{\varepsilon i(\nu)}(M_{\varepsilon i}, W_{\varepsilon i}, i_{\varepsilon i}) \cdot \ln \phi_{\varepsilon i(\nu)}(M_{\varepsilon i}, W_{\varepsilon i}, i_{\varepsilon i}) \ . \tag{3.11}$$

This signal entropy is the one produced by the signal dynamics, and it reaches zero only if we insert a product of δ functions for ϕ which draws upon exact knowledge of the solutions of the signal equations. As this knowledge is not available for larger nets, the frequency distribution must be empirically constructed by means of equating „temporal mean and spatial mean“[3]. The measurement process then determines the exact value of $S_0 > 0$.

3.3 Determination of the Frequency Distribution

Let us first suppose that knowledge exists about the frequency of occurrence of a partition and its mean energy, ie. the signal solutions can not be computed any longer. Remembering the interrelationship between net size and mean life time of partitions (Figure 2.1) it becomes clear that we must admit to little lack of knowledge only. Nevertheless, it is reasonable to look for that distribution which satisfies the side conditions (3.8, 3.10) and forces the entropy to become maximal.

By calculating the functional derivative of

$$\begin{aligned} S_0 &+ \sum_{\varepsilon=0}^{N}\sum_{i=1}^{\binom{N}{\varepsilon}}\sum_{\nu=1}^{B_{\varepsilon i}}\lambda_{\varepsilon i(\nu)}\left(P_{\varepsilon i(\nu)} - \int di_{\varepsilon i}\, dM_{\varepsilon i}\, dW_{\varepsilon i}\, \psi_{\varepsilon i(\nu)}(M_{\varepsilon i}, W_{\varepsilon i}, i_{\varepsilon i})\right) \\ &+ \sum_{\varepsilon=0}^{N}\sum_{i=1}^{\binom{N}{\varepsilon}}\sum_{\nu=1}^{B_{\varepsilon i}}\beta_{\varepsilon i(\nu)}\left(H_{\varepsilon i(\nu)} - \int di_{\varepsilon i}\, dM_{\varepsilon i}\, dW_{\varepsilon i}\, \psi_{\varepsilon i(\nu)}(M_{\varepsilon i}, W_{\varepsilon i}, i_{\varepsilon i}) \cdot h_{\varepsilon i(\nu)}\right) \end{aligned} \tag{3.12}$$

we obtain as expressions for the distribution and the entropy, respectively:

$$\begin{aligned} \psi_{\varepsilon i(\nu)}(M_{\varepsilon i}, W_{\varepsilon i}, i_{\varepsilon i}) &= \exp\left[-\lambda_{\varepsilon i(\nu)} + 1 - \beta_{\varepsilon i(\nu)} \cdot h_{\varepsilon i(\nu)}(M_{\varepsilon i}, W_{\varepsilon i}, i_{\varepsilon i})\right] \ , \\ S_1 &= \sum_{\varepsilon=0}^{N}\sum_{i=1}^{\binom{N}{\varepsilon}}\sum_{\nu=1}^{B_{\varepsilon i}}\left[P_{\varepsilon i(\nu)} \cdot (\lambda_{\varepsilon i(\nu)} - 1) + \beta_{\varepsilon i(\nu)} \cdot H_{\varepsilon i(\nu)}\right] \ . \end{aligned} \tag{3.13}$$

With $\xi_{\varepsilon i(\nu)} := \lambda_{\varepsilon i(\nu)} - 1$ one calculates:

[3] In Mathematics the frequency distribution is called density function, in Information Theory it is called distribution function or joint distribution. From now on we will be using the term „distribution“.

$$\xi_{\varepsilon i(\nu)} = -\ln P_{\varepsilon i(\nu)} + \ln \int di_{\varepsilon i} dM_{\varepsilon i} dW_{\varepsilon i} \exp\left[-\beta_{\varepsilon i(\nu)} \cdot h_{\varepsilon i(\nu)}(M_{\varepsilon i}, W_{\varepsilon i}, i_{\varepsilon i})\right] \ .$$

Therefore the entropy reads:

$$S_1 := \sum_{\varepsilon=0}^{N} \binom{N}{\varepsilon} \sum_{i=1}^{B_{\varepsilon i}} \sum_{\nu=1} \beta_{\varepsilon i(\nu)} \cdot H_{\varepsilon i(\nu)} - \sum_{\varepsilon=0}^{N} \binom{N}{\varepsilon} \sum_{i=1}^{B_{\varepsilon i}} \sum_{\nu=1} P_{\varepsilon i(\nu)}$$
$$\cdot \left[\ln P_{\varepsilon i(\nu)} - \ln \int di_{\varepsilon i} dM_{\varepsilon i} dW_{\varepsilon i} \exp\left[-\beta_{\varepsilon i(\nu)} \cdot h_{\varepsilon i(\nu)}(M_{\varepsilon i}, W_{\varepsilon i}, i_{\varepsilon i})\right]\right] \ . \tag{3.14}$$

Because knowledge has been surrendered as compared with (3.11) it holds that: $S_0 < S_1$. As will be shown in chapter 5 and chapter 7, the case under consideration is relevant for neural nets with dynamic synapses. Then, neurons can synchronise, and the state space becomes observable.

In the next step we give up our knowledge about durations $\tau_{\varepsilon i(\nu)}$, the multiplicities $B_{\varepsilon i}$, and the mean energies $H_{\varepsilon i(\nu)}$. Hence, it is no longer known from which particular partition a partition stems or into which one it will transit. This means that in (3.8) the ν-sum must be replaced by a spatial one. Now it is necessary to remember that the distribution $\phi_{\varepsilon i(\nu)}$ depends on the start time and the duration or end time of a partition. As the $t_{\varepsilon i(\nu)}$ are unique functions of the initial values $a^0_{\varepsilon i} := a_{\varepsilon i(\nu)}(t_{\varepsilon i(\nu)})$ and $\Delta^0_{\varepsilon i} := \Delta_{\varepsilon i(\nu)}(t_{\varepsilon i(\nu)})$, but all knowledge about them is given up, the ν-summation involves the integration over all initial values. Likewise one could expect that the end time $t_{\varepsilon i(\nu)} + \tau_{\varepsilon i(\nu)}$ would entail the integration over all end values $a^1_{\varepsilon i} := a_{\varepsilon i(\nu)}(t_{\varepsilon i(\nu)} + \tau_{\varepsilon i(\nu)})$ and $\Delta^1_{\varepsilon i} := \Delta_{\varepsilon i(\nu)}(t_{\varepsilon i(\nu)} + \tau_{\varepsilon i(\nu)})$. However, this does not hold true, for the end values of the membrane potentials result from their initial values and the M values at the end of a partition, which have been taken care of already by the M-integration. Similarly for the end values $\Delta_{\varepsilon i(\nu)}(t_{\varepsilon i(\nu)} + \tau_{\varepsilon i(\nu)})$ according to (2.2b, d), we obtain the expression $\Delta^0_{\varepsilon i} + t_d - \Delta_x(t_{\varepsilon i(\nu)})$, with x as a deceasing pulse neuron, or we obtain $\Delta_{\varepsilon i(\nu)}(t_{\varepsilon i(\nu)} + \tau_{\varepsilon i(\nu)})$. Here $\tau_{\varepsilon i(\nu)}$ designates the time the receiving neuron needed to arrive at the threshold as first neuron, before any other transmitting neurons could have a chance to decease. Hence, the end values represent the information about the change of a partition. Because we do not possess this information the ν-sum requires an additional summation for the end values over all partitions $\{\varepsilon' i'\}$ which an individual partition may transit into.

This way we obtain:

$$H(t_0, T) = \sum_{\varepsilon=0}^{N} \binom{N}{\varepsilon} \sum_{i=1} \int di_{\varepsilon i} da^0_{\varepsilon i} d\Delta^0_{\varepsilon i} dM_{\varepsilon i} dW_{\varepsilon i}$$
$$\sum_{\varepsilon'=0}^{N} \binom{N}{\varepsilon'} \sum_{i'=1} f_{\varepsilon i \varepsilon' i'}(M_{\varepsilon i}, W_{\varepsilon i}, a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}, i_{\varepsilon i}) \cdot h_{\varepsilon i}(M_{\varepsilon i}, W_{\varepsilon i}, a^0_{\varepsilon i}, i_{\varepsilon i}) \ . \tag{3.15a}$$

Since the signal energy does not contain any information about the transitions of partitions the inner sum can be taken care of by the redefinition of the distribution as:

$$H(t_0,T)=\sum_{\varepsilon=0}^{N}\sum_{i=1}^{\binom{N}{\varepsilon}}\int di_{\varepsilon i}da^0_{\varepsilon i}d\Delta^0_{\varepsilon i}dM_{\varepsilon i}dW_{\varepsilon i} \\ f_{\varepsilon i}(M_{\varepsilon i},W_{\varepsilon i},a^0_{\varepsilon i},\Delta^0_{\varepsilon i},i_{\varepsilon i})\cdot h_{\varepsilon i}(M_{\varepsilon i},W_{\varepsilon i},a^0_{\varepsilon i},i_{\varepsilon i})\ . \tag{3.15b}$$

The new distribution must be normalized,

$$\sum_{\varepsilon=0}^{N}\sum_{i=1}^{\binom{N}{\varepsilon}}\int di_{\varepsilon i}da^0_{\varepsilon i}d\Delta^0_{\varepsilon i}dM_{\varepsilon i}dW_{\varepsilon i}f_{\varepsilon i}(M_{\varepsilon i},W_{\varepsilon i},a^0_{\varepsilon i},\Delta^0_{\varepsilon i},i_{\varepsilon i})=1\ , \tag{3.16}$$

and the entropy

$$-\sum_{\varepsilon=0}^{N}\sum_{i=1}^{\binom{N}{\varepsilon}}\int di_{\varepsilon i}da^0_{\varepsilon i}d\Delta^0_{\varepsilon i}dM_{\varepsilon i}dW_{\varepsilon i} \\ f_{\varepsilon i}(M_{\varepsilon i},W_{\varepsilon i},a^0_{\varepsilon i},\Delta^0_{\varepsilon i},i_{\varepsilon i})\cdot\ln f_{\varepsilon i}(M_{\varepsilon i},W_{\varepsilon i},a^0_{\varepsilon i},\Delta^0_{\varepsilon i},i_{\varepsilon i}):=S_2 \tag{3.17}$$

must become extremal, taking into account the side conditions (3.15b, 3.16). After some calculations have been performed we get the results:

$$f_{\varepsilon i}(M_{\varepsilon i},W_{\varepsilon i},a^0_{\varepsilon i},\Delta^0_{\varepsilon i},i_{\varepsilon i})=\exp[-\xi-\beta\cdot h_{\varepsilon i}(M_{\varepsilon i},W_{\varepsilon i},a^0_{\varepsilon i},i_{\varepsilon i})]\ , \\ S_2=\xi+\beta\cdot H\,,\quad \xi:=\ln Z\,,\quad Z=\sum_{\varepsilon=0}^{N}\sum_{i=1}^{\binom{N}{\varepsilon}}Z_{\varepsilon i}\,, \\ Z_{\varepsilon i}=\int di_{\varepsilon i}da^0_{\varepsilon i}d\Delta^0_{\varepsilon i}dM_{\varepsilon i}dW_{\varepsilon i}\ \exp[-\beta\cdot h_{\varepsilon i}(M_{\varepsilon i},W_{\varepsilon i},a^0_{\varepsilon i},i_{\varepsilon i})]\ . \tag{3.18}$$

Overall the identity

$$H=-\frac{\partial\xi}{\partial\beta}=-\frac{1}{Z}\cdot\frac{\partial Z}{\partial\beta} \tag{3.19}$$

gives the value for β, so that Z and therewith ξ and S_2 obtain fixed values. (H is given by (3.7) or assumed to be measurable by experiment).

It is convenient to use the following abbreviations for temporal and spatial mean values, respectively, of an arbitrary function $R(M,W,a^0,\Delta^0,i)$:

$$\langle R\rangle:=\sum_{\varepsilon=0}^{N}\sum_{i=1}^{\binom{N}{\varepsilon}}\langle R_{\varepsilon i}\rangle:=\sum_{\varepsilon=0}^{N}\sum_{i=1}^{\binom{N}{\varepsilon}}\int di_{\varepsilon i}da^0_{\varepsilon i}d\Delta^0_{\varepsilon i}dM_{\varepsilon i}dW_{\varepsilon i} \\ f_{\varepsilon i}(M_{\varepsilon i},W_{\varepsilon i},a^0_{\varepsilon i},\Delta^0_{\varepsilon i},i_{\varepsilon i})\cdot R_{\varepsilon i}(M_{\varepsilon i},W_{\varepsilon i},a^0_{\varepsilon i},\Delta^0_{\varepsilon i},i_{\varepsilon i})\ . \tag{3.20}$$

With it and (3.19) it results for the variation of the signal energy:

$$[\Delta h]^2 := \langle h^2 \rangle - \langle h \rangle^2 = \frac{\partial^2 \xi}{\partial \beta^2} \ . \tag{3.21}$$

The expression $\Delta h / \langle h \rangle$ should tend to vanish with growing net size.

With (3.16) and (3.10) the probability of a partition $\{\varepsilon i\}$ reads[4]:

$$P_{\varepsilon i} = \int di_{\varepsilon i} da^0_{\varepsilon i} d\Delta^0_{\varepsilon i} dM_{\varepsilon i} dW_{\varepsilon i} f_{\varepsilon i}(M_{\varepsilon i}, W_{\varepsilon i}, a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}, i_{\varepsilon i}) = \frac{Z_{\varepsilon i}}{Z} = \frac{\tau_{\varepsilon i}}{T} \ . \tag{3.22}$$

Further, the probability that a neuron n is in its accumulation and transmitting phase, respectively, reads:

$$P_{An} := \sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} P_{\varepsilon i} \cdot A_{\varepsilon i,n} \ , \ P_{Sn} := \sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} P_{\varepsilon i} \cdot S_{\varepsilon i,n} \ . \tag{3.23}$$

Here $A_{\varepsilon i,n}$ and $S_{\varepsilon i,n}$, resp., takes on the value 1 if n is a receiving and transmitting neuron, resp., in the partition $\{\varepsilon i\}$, otherwise 0. This way the pulse rate of a neuron or the mean pulse rate of (an ensemble of) a net can be tied to the theoretical description of the net dynamics. For let $P_{An} \cdot T$ designate the sum of accumulation times of neuron n in the observation period T. Then the number of pulses produced by the neuron n in this period of time accumulates to $(1 - P_{An}) \cdot T / t_d$. Therefore, the pulse rate F_n of a neuron n or the mean pulse rate are given by:

$$F_n = \frac{P_{Sn}}{t_d} = \frac{1}{t_d} \sum_{\varepsilon=0}^{N-1} \sum_{i=1}^{\binom{N}{\varepsilon}} \frac{Z_{\varepsilon i}}{Z} \cdot S_{\varepsilon i,n} \quad , \quad F = \frac{1}{t_d} \cdot \frac{1}{N} \sum_{\varepsilon=0}^{N-1} (N - \varepsilon) \sum_{i=1}^{\binom{N}{\varepsilon}} \frac{Z_{\varepsilon i}}{Z} \ . \tag{3.24a}$$

The mean pulse rate can be written shorter with the help of (3.24a):

$$F = \frac{1}{t_d} \cdot \frac{1}{N} \cdot \langle \text{number of transmitter} =: \#s \rangle = \frac{1}{N} \cdot \frac{\partial \xi}{\partial t_d} \ , \tag{3.24b}$$

where we determined that the integrals over Δ^0 (3.18) in the partition $\{\varepsilon i\}$ take on the value $t_d^{(N-\varepsilon)}$.

Defining the variation of the mean pulse rate according to (3.21), we obtain:

[4] This quantity, the integral of the distribution over a partition, was called in chapter 2 already the frequency distribution, or distribution for short. We hope that this double use is resolved by the context.

$$
\begin{aligned}
[\Delta F]^2 &:= \frac{1}{t_d^2}\cdot\frac{1}{N^2}(\langle \#s^2\rangle - \langle \#s\rangle^2) \\
&= \frac{1}{t_d^2}\cdot\frac{1}{N^2}\left[\sum_{\varepsilon=0}^{N-1}(N-\varepsilon)^2\sum_{i=1}^{\binom{N}{\varepsilon}}\frac{Z_{\varepsilon i}}{Z} - \left(\sum_{\varepsilon=0}^{N-1}(N-\varepsilon)\sum_{i=1}^{\binom{N}{\varepsilon}}\frac{Z_{\varepsilon i}}{Z}\right)^2\right] \\
&= \frac{1}{t_d}\cdot\frac{1}{N^2}\cdot\frac{\partial\langle \#s\rangle}{\partial t_d} = \frac{1}{t_d}\cdot\frac{1}{N^2}\cdot\frac{\partial}{\partial t_d}\left[t_d\cdot\frac{\partial\xi}{\partial t_d}\right] \\
&= \frac{1}{t_d}\cdot\frac{1}{N^2}\cdot\left[\frac{\partial\xi}{\partial t_d} + t_d\cdot\frac{\partial^2\xi}{\partial t_d^2}\right] \\
&= \frac{F}{t_d\cdot N} + \frac{1}{N}\cdot\frac{\partial F}{\partial t_d}\ .
\end{aligned}
\tag{3.24c}
$$

As the pulse rate is limited when the net is in the cold operating range for large nets, this results in a vanishing variation of the mean pulse rate for large neuron numbers. This is in agreement with the experiments (Figure 2.14 right).

Other useful relations result when the derivative of the entropy is calculated with respect to the (constant) input i_e and the temporal mean $\hat{i}_e$ of the input signals $i_e(t)$, respectively. Let X_e designate any of the constant input signals or their mean value. Since β and ξ are both a function of $H(X_e), X_e$ it follows that the entropy is a function of $H(X_e), X_e$. Then, the differential of S reads:

$$
\begin{aligned}
dS &= d\xi + H\cdot d\beta + \beta\cdot dH = \sum_e\left(\frac{\partial\xi}{\partial X_e} + H\cdot\frac{\partial\beta}{\partial X_e}\right)\cdot dX_e + \beta\cdot dH \\
&= \sum_e\left\{-H\cdot\frac{\partial\beta}{\partial X_e} + A(X_e) + H\cdot\frac{\partial\beta}{\partial X_e}\right\}\cdot dX_e + \beta\cdot dH \\
&= \sum_e A(X_e)\cdot dX_e + \beta\cdot dH \qquad \Leftrightarrow \qquad \frac{\partial S}{\partial X_e} = A(X_e)\ ,\ \frac{\partial S}{\partial H} = \beta \\
&\Leftrightarrow \qquad S = \int ds = \int dX_e\left\{\frac{\partial S}{\partial X_e} + \beta(X_e)\cdot\frac{\partial H}{\partial X_e}\right\}\ .
\end{aligned}
\tag{3.25}
$$

$A(X_e)$ is an abbreviation for the expression:

$$
\begin{aligned}
&\sum_{\varepsilon=1}^{N}\sum_{i=1}^{\binom{N}{\varepsilon}}\int da^0_{\varepsilon i}d\Delta^0_{\varepsilon i}dM_{\varepsilon i}dW_{\varepsilon i} \\
&\qquad\frac{\partial}{\partial X_e}\left\{\int di_{\varepsilon i}\exp[-\xi-\beta\cdot h_{\varepsilon i}(M_{\varepsilon i},W_{\varepsilon i},a^0_{\varepsilon i},i_{\varepsilon i})]\right\}\Big|_{\xi,\beta\text{ fixed}} \\
&= \sum_{\varepsilon=1}^{N}\sum_{i=1}^{\binom{N}{\varepsilon}}\frac{1}{Z}\frac{\partial Z_{\varepsilon i}}{\partial X_e}\Big|_{\beta\text{ fixed}}\ .
\end{aligned}
\tag{3.26}
$$

The last line of (3.25) says that the partial derivatives $\partial S/\partial X_e$ and $\partial H/\partial X_e$ as well as β must be computed or measured as function of the input signals in order to know the entropy.

The calculation of (3.26) is far more elaborate for the general case of the dynamic synapses than for the constant ones. Because the „constant case" is of importance for the application we discontinue the explanation of the theory of the general case and apply the theory to constant synapses in the next chapter. The test of the „constant theory" is also done in this chapter on the basis of the experiments of chapter 2.

3.4 Summary

The computation of the quantities of entropy and pulse rate found in the experiments are based on two theoretical steps. Firstly the original signal equations for neurons and synapses are tied uniquely to the signal energy of the net by means of a partial differential equation of the first order. The signal energy is the sum of the energy stored in synapses and membranes. Secondly, the temporal mean of the signal energy is transformed into a spatial mean on the basis of a distribution function. The latter counts how often each set of synaptic and neural values occurs for each partition during the observation period of the net. For small nets the distribution can be determined experimentally. For larger nets the distribution is found to be the solution of an optimisation problem: We are aiming for that distribution which gives the measured mean signal energy as an expectation value of the signal energy and which allows the entropy to become maximal. The latter is a reflection of the fact that the signal equations are well established, but not solvable or measurable[5].

The entropy turns out to be the basic quantity from which the pulse rate of any neuron, the net, and the ensemble or the dependency of the entropy from the input signals can be derived, be it constant or pulse signals.

[5] The explanation of the context of signal equations and entropy as presented in this chapter will remind some readers of *thermodynamics*. There the energy function is the Hamiltonian of a system of particles which on the one hand creates the Hamiltonian equations (the equivalent of our signal equations) by means of the Hamilton-Jacobi partial differential equation ($micro - dynamics$), and which on the other hand determines the analytic structure of the distribution function from which the entropy is calculated ($macro - dynamics$). A simple transfer of the underlying structure was not possible, however, for the finite life time of a partition in the cases of *gases* finds it counterpart only when inelastic collisions of the molecules would have to be considered. I do not know of any dealings with such a case, which has resulted in the reflection on the basic equivalence of temporal and spatial means.

4

Macro-Dynamics of Nets with Constant Synapses

Ulrich Ramacher and Jens Harnisch

For constant synapses it is sufficient to consider the partial differential equation

$$\frac{\partial J_{\varepsilon i}(M_{\varepsilon i}, i_{\varepsilon i}(t))}{\partial t} + h_{\varepsilon i}(M_{\varepsilon i}, W_{\varepsilon i}, i_{\varepsilon i}(t)) = 0 \ , \tag{4.1}$$

with the signal energy given by

$$h_{\varepsilon i}(W_{\varepsilon i}, i_{\varepsilon i}(t)) = \frac{1}{2} \sum_{e \in E_{\varepsilon i}} \left\{ \sum_{s \in S_{\varepsilon i} \cap N_e} W_{es}^2 \cdot x_s + W_{e0}^2 \cdot i_e(t) \right\} \ .$$

Then the characteristic equations (3.5b) read:

$$\begin{aligned} \frac{dM_{es}}{dt} = W_{es} \cdot x_s \ , \ \frac{dM_{e0}}{dt} = W_{e0} \cdot i_e \ , \quad \frac{dW_{es}}{dt} = 0 \ , \ \frac{dW_{e0}}{dt} = 0, \\ \frac{d(\partial J_{\varepsilon i}/\partial t)}{dt} = - \sum_{e \in E_{\varepsilon i}} \frac{W_{e0}^2}{2} \cdot \frac{di_e}{dt} \ , \\ \frac{dJ_{\varepsilon i}}{dt} = \frac{1}{2} \sum_{e \in E_{\varepsilon i}} \left\{ \sum_{s \in S_{\varepsilon i} \cap N_e} W_{es}^2 \cdot x_s + W_{e0}^2 \cdot i_e(t) \right\} = h_{\varepsilon i}(W_{\varepsilon i}, i_{\varepsilon i}(t)) \ . \end{aligned} \tag{4.2}$$

Hence, we have to deal with a system, the macro-dynamics of which depends only on the initial values of each partition, the external signals, and a fixed set of constant synaptic values.

4.1 Known Synapses

First, we assume that the values of the synapses are known. The knowledge about the W- and M-functions is taken care of during the change from temporal to spatial mean:

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 59–81, 2010.
© Springer-Verlag Berlin Heidelberg 2010

$$\wedge\, \varepsilon i \wedge e \in E_{\varepsilon i} \wedge s \in S_{\varepsilon i} \cap N_e \;:\; \int dW'_{es}\delta(W'_{es} - W_{es}) = 1 \;, \int dM_{es}\phi(M_{es}) = 1$$
$$\wedge\, \varepsilon i \wedge e \in E_{\varepsilon i} \;:\; \int dW'_{e0}\delta(W'_{e0} - W_{e0}) = 1 \;, \int dM_{e0}\phi(M_{e0}) = 1 \;.$$

We obtain for the spatial mean of the energy:

$$\sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} \int di_{\varepsilon i} da^0_{\varepsilon i} d\Delta^0_{\varepsilon i}\, f_{\varepsilon i}(a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}, i_{\varepsilon i}; W_{\varepsilon i}) \cdot h_{\varepsilon i}(i_{\varepsilon i}; W_{\varepsilon i}) = H \;, \tag{4.3a}$$

where f designates again the unknown distribution. The semicolon separates parameters from variables.

The distribution must be normalized according to

$$\sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} \int di_{\varepsilon i} da^0_{\varepsilon i} d\Delta^0_{\varepsilon i}\, f_{\varepsilon i}(a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}, i_{\varepsilon i}; W_{\varepsilon i}) = 1, \tag{4.3b}$$

and the entropy

$$-\sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} \int di_{\varepsilon i} da^0_{\varepsilon i} d\Delta^0_{\varepsilon i}\, f_{\varepsilon i}(a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}, i_{\varepsilon i}; W_{\varepsilon i}) \cdot \ln f_{\varepsilon i}(a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}, i_{\varepsilon i}; W_{\varepsilon i}) := S$$

must become maximal while fulfilling the side conditions (4.3a,b).

Then this results in:

$$f_{\varepsilon i}(a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}, i_{\varepsilon i}; W_{\varepsilon i}) = \exp[-\xi - \beta \cdot h_{\varepsilon i}(i_{\varepsilon i}; W_{\varepsilon i})] \;,$$
$$S = \xi + \beta \cdot H \;, \quad \xi := \ln Z \;, \quad Z = \sum_{\varepsilon=1}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} Z_{\varepsilon i} \;,$$
$$\begin{aligned} Z_{\varepsilon i} &= \int di_{\varepsilon i} da^0_{\varepsilon i} d\Delta^0_{\varepsilon i}\, \exp[-\beta \cdot h_{\varepsilon i}(i_{\varepsilon i}; W_{\varepsilon i})] \\ &= \prod_{s \in S_{\varepsilon i}} \left(\int_0^{t_d} d\Delta^0_s \right) \cdot \prod_{e \in E_{\varepsilon i}} \left(\int_0^{\theta} da^0_e \right) \cdot \int di_{\varepsilon i} \exp[-\beta \cdot h_{\varepsilon i}(i_{\varepsilon i}; W_{\varepsilon i})] \\ &= t_d^{N-\varepsilon} \cdot \theta^{\varepsilon} \cdot \int di_{\varepsilon i} \exp[-\beta \cdot h_{\varepsilon i}(i_{\varepsilon i}; W_{\varepsilon i})] \;. \end{aligned} \tag{4.4}$$

Again, the identity

$$H = -\frac{\partial \xi}{\partial \beta} = -\frac{1}{Z} \cdot \frac{\partial Z}{\partial \beta} \tag{4.5a}$$

delivers the value for β so that finally Z, ξ and S will have fixed values. Analogously, we proceed with all further explanations developed in chapter 3.

Let us consider first the case of constant input signals, and let $\hat{i}_e$ denote the temporal average of the e-th input signal.

A. Constant Input Signals: $\hat{i}_e = i_e$

Henceforth, the energy depends on parameters only, and the distribution f is a *delta* function with respect to i_e. From (4.4) the result for $Z_{\varepsilon i}$ follows:

$$Z_{\varepsilon i} = t_d^{N-\varepsilon} \cdot \theta^{\varepsilon} \cdot e^{-\beta \cdot h_{\varepsilon i}(i_{\varepsilon i}, W_{\varepsilon i})} \ . \tag{4.5b}$$

From this and with (4.5a) we calculate:

$$H = \sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} \langle h_{\varepsilon i} \rangle = \sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} \frac{Z_{\varepsilon i}}{Z} \cdot h_{\varepsilon i}(i_{\varepsilon i}, W_{\varepsilon i}) = \sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} P_{\varepsilon i} \cdot h_{\varepsilon i}(i_{\varepsilon i}, W_{\varepsilon i}) \ . \tag{4.5c}$$

This is a very plausible result.

While we insert the measured value for H, the value of β can be calculated so that $S = \xi + \beta \cdot H$ is on hand in calculable form. S can be written also:

$$\begin{aligned} S &= \sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} P_{\varepsilon i} \cdot [\ln Z + \beta \cdot h_{\varepsilon i}(i_{\varepsilon i}, W_{\varepsilon i})] \\ &= \sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} P_{\varepsilon i} \cdot [-\ln P_{\varepsilon i} + \ln Z_{\varepsilon i} + \beta \cdot h_{\varepsilon i}(i_{\varepsilon i}, W_{\varepsilon i})] \\ &= -\sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} P_{\varepsilon i} \cdot [\ln P_{\varepsilon i} - (N-\epsilon) \cdot \ln t_d - \epsilon \cdot \ln \theta] \ . \end{aligned} \tag{4.5d}$$

Here we must remember that the first summand only of the bracket in (4.5d) contributes to the experimental entropy in chapter 2.

Finally, with (3.26) we obtain for $\partial S / \partial i_e$:

$$\frac{\partial S}{\partial i_e} = -\beta \sum_{\varepsilon=1}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} P_{\varepsilon i} \cdot \frac{W_{e0}^2 \cdot A_{\varepsilon i, e}}{2} = -\beta \cdot P_{Ae} \cdot \frac{W_{e0}^2}{2} \ , \tag{4.6a}$$

and for $\partial S / \partial x_s$:

$$\frac{\partial S}{\partial x_s} = -\beta \sum_{\varepsilon=1}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} P_{\varepsilon i} \cdot \sum_{e \in E_{\varepsilon i}} \frac{W_{es}^2 \cdot \chi_{\varepsilon i}(e, s)}{2} \ , \tag{4.6b}$$

where the quantity $\chi_{\varepsilon i,e}(e,s)$ has the value 1 if $s \in \{S_{\varepsilon i} \cap N_e\}$ holds, otherwise zero. Note that this relationship - just as (4.6a) - holds for fixed signal energy H (cf. 3.25).

B. Pulses as Input Signals: $\hat{i}_e := 1/T \cdot \int i_e(t)dt = x \cdot F_e \cdot t_d$

Because an important case for the application is the one in which pulses are delivered to a net as input signals, we now assume that only the distributions $\omega(i_e)$ of the values 0 and 1 for the pulses sent by some other net are known, but not the times of arrival of the pulses at the input neurons. As the occurrence of the pulses at each input neuron are assumed to be independent from the dynamics of the receiving net, the distribution from (3.18) reads:

$$f_{\varepsilon i}(M_{\varepsilon i}, W_{\varepsilon i}, a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}, i_{\varepsilon i}) = \phi_{\varepsilon i}(M_{\varepsilon i}, W_{\varepsilon i}, a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}; \omega(i_{\varepsilon i})) \cdot \prod_{e \in E_{\varepsilon i}} \omega(i_{\varepsilon i}) \ ,$$

and from the normalization condition for each $\omega(i_e)$,

$$1 = \int di_e \ \omega(i_e) \ ,$$

the normalization condition for $\phi_{\varepsilon i}$ results:

$$1 = \int da^0_{\varepsilon i} d\Delta^0_{\varepsilon i} dM_{\varepsilon i} dW_{\varepsilon i} \, \phi_{\varepsilon i}(M_{\varepsilon i}, W_{\varepsilon i}, a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}; \omega(i_{\varepsilon i})) \ .$$

With the decomposition $h_{\varepsilon i} = h^1_{\varepsilon i}(W_{\varepsilon i}) + h^0_{\varepsilon i}(W^0_{\varepsilon i}, i_{\varepsilon i})$ the calculation of the mean of the energy according to (3.15b) produces in our case of the known synapses (and with the reusing of f as symbol of the remaining distribution):

$$\begin{aligned} H = & \sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} \int da^0_{\varepsilon i} d\Delta^0_{\varepsilon i} \, f_{\varepsilon i}(a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}; \omega(i_{\varepsilon i})) \\ & \cdot \left[h^1_{\varepsilon i}(W_{\varepsilon i}) + \sum_{e \in E_{\varepsilon i}} \int di_e \ \omega(i_e) \cdot \frac{W^2_{e0} \cdot i_e}{2} \right] \ . \end{aligned} \tag{4.7}$$

Let $\{t_{en}\}_{n \in \mathbf{N}}$ designate the (unknown) time points of the occurrence of a pulse at the e-th input neuron. The input signal can be written then:

$$i_e(t) = x \cdot \chi_e(t) \ , \text{with } \chi_e(t) = \begin{cases} 1 & \text{for } t_{en} < t < t_{en} + 1ms \\ 0 & \text{otherwise} \ . \end{cases}$$

If we use this definition for the transformation of a temporal mean into a spatial one (3.15b), the integration over i_e results in a single non-zero term:

$$H = \sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} \int da^0_{\varepsilon i} d\Delta^0_{\varepsilon i}\, f_{\varepsilon i}(a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}; \omega(i_{\varepsilon i})) \cdot \left[h^1_{\varepsilon i}(W_{\varepsilon i}) + \sum_{e \in E_{\varepsilon i}} \frac{\omega(i_e = x) \cdot W^2_{e0} \cdot i_e}{2} \right] . \tag{4.8a}$$

The entropy

$$S = -\sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} \int da^0_{\varepsilon i} d\Delta^0_{\varepsilon i}\, f_{\varepsilon i}(a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}; \omega(i_{\varepsilon i})) \cdot \ln f_{\varepsilon i}(a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}; \omega(i_{\varepsilon i})) \tag{4.8b}$$

will become maximal for the distribution:

$$f_{\varepsilon i}(a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}; \omega(i_{\varepsilon i})) = \exp[-\xi - \beta \cdot h_{\varepsilon i}(; W_{\varepsilon i}, \omega_{\varepsilon i})] ,$$
$$\text{with} \quad h_{\varepsilon i}(; W_{\varepsilon i}, \omega_{\varepsilon i}) := h^1_{\varepsilon i}(W_{\varepsilon i}) + \sum_{e \in E_{\varepsilon i}} \frac{\omega_e(i_e = x) \cdot W^2_{e0} \cdot i_e}{2} .$$

Since the probability $\omega_e(i_e = x)$ is identical with the product $F_e \cdot t_d$ of the pulse rate and the duration of a pulse, and the temporal mean $\hat{i}_e$ of a pulse signal i_e at the input side is given by $F_e \cdot t_d \cdot x$, we need only to replace i by $\hat{i}$. In particular, the result for $Z_{\varepsilon i}$ is:

$$Z_{\varepsilon i} = t_d^{N-\varepsilon} \cdot \theta^{\varepsilon} \cdot e^{-\beta \cdot h_{\varepsilon i}(; W_{\varepsilon i}, \hat{i}_{\varepsilon i})} , \tag{4.8c}$$

and for H it results with (4.8a):

$$H = \sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} \int da^0_{\varepsilon i} d\Delta^0_{\varepsilon i}\, f_{\varepsilon i}(a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}; \hat{i}_{\varepsilon i}) \cdot h_{\varepsilon i}(; W_{\varepsilon i}, \hat{i}_{\varepsilon i}) . \tag{4.8d}$$

After replacing i by $\hat{i}$ (4.6a,b) will hold thereafter.

4.2 Known Distribution of Synapses

Next, let the knowledge about the distribution $v(W)$ of the synaptic values be available (typical for the electronic realisation of neural nets). In this case we are dealing with an ensemble of nets which differ by possible initial values and the set of representative synapses chosen. In this book we focus on an equipartition given by the mean values W and w, resp., and the intervals $[W - \Delta W, W + \Delta W]$ and $[w - \Delta w, w + \Delta w]$, resp. As part of the distribution

f is known already - just as was the case for pulses as input signals - we can transfer the analysis from section 4.1 and obtain:

$$\wedge\, \varepsilon i \wedge e \in E_{\varepsilon i} \wedge s \in S_{\varepsilon i} \cap N_e \;:\; \int dW_{es} v(W_{es}) = 1 \;,\; \int dM_{es} \phi(M_{es}) = 1$$
$$\wedge\, \varepsilon i \wedge e \in E_{\varepsilon i} \;:\; \int dW_{e0} v(W_{e0}) = 1 \;,\; \int dM_{e0} \phi(M_{e0}) = 1 \;.$$

Using again the symbol f for the remaining distribution, it remains true that:

$$\begin{aligned}
&\sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} \int di_{\varepsilon i} da^0_{\varepsilon i} d\Delta^0_{\varepsilon i}\, f_{\varepsilon i}(a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}, i_{\varepsilon i}; W) = 1 \;, \\
&\sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} \int di_{\varepsilon i} da^0_{\varepsilon i} d\Delta^0_{\varepsilon i}\, f_{\varepsilon i}(a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}, i_{\varepsilon i}; W) \int v_{\varepsilon i}(W_{\varepsilon i}) \cdot h_{\varepsilon i}(W_{\varepsilon i}, i_{\varepsilon i}) = H \;, \\
-&\sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} \int di_{\varepsilon i} da^0_{\varepsilon i} d\Delta^0_{\varepsilon i}\, f_{\varepsilon i}(M_{\varepsilon i}, a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}, i_{\varepsilon i}; W) \cdot \ln f_{\varepsilon i} =: S \;,
\end{aligned} \tag{4.9a}$$

with the integrals reaching from $W - \Delta W$ to $W + \Delta W$ and from $w - \Delta w$ to $w + \Delta w$, respectively. For ease of readability, the parameterization of the remaining distribution f by the mean values W and w, resp., and variations ΔW and Δw, resp., are not shown, but indicated by the single notation of W after the semicolon. Similarly for the energy h. The calculation of the W-integrals produces:

$$\begin{aligned}
h_{\varepsilon i}(i_{\varepsilon i}; W) &= \int dW_{\varepsilon i} v_{\varepsilon i}(W_{\varepsilon i}) \cdot h_{\varepsilon i}(W_{\varepsilon i}, i_{\varepsilon i}) \\
&= x \cdot \sum_{e=1}^{\varepsilon} s_{e\varepsilon i} \cdot \left[\frac{W^2}{2} + \frac{(\Delta W)^2}{6} \right] + \left(\sum_{e=1}^{\varepsilon} i_e \right) \left[\frac{w^2}{2} + \frac{(\Delta w)^2}{6} \right] \;.
\end{aligned} \tag{4.9b}$$

Here $s_{e\varepsilon i}$ designates the number of transmitting neurons in the partition $\{\varepsilon i\}$ which are connected to the e-th receiving neuron. This way, we receive as expression for the energy H:

$$H = \sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} \int da^0_{\varepsilon i} d\Delta^0_{\varepsilon i}\, f_{\varepsilon i}(a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}, i_{\varepsilon i}; W) \cdot h_{\varepsilon i}(i_{\varepsilon i}; W) \;. \tag{4.9c}$$

By calculating the functional derivative we obtain:

$$f_{\varepsilon i}(a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}, i_{\varepsilon i}; W) = \exp[-\xi - \beta \cdot h_{\varepsilon i}(i_{\varepsilon i}; W)] \ , S = \xi + \beta \cdot H \ , \xi := \ln Z \ ,$$
$$Z = \sum_{\varepsilon=1}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} Z_{\varepsilon i} \ , \quad Z_{\varepsilon i} = t_d^{N-\varepsilon} \cdot \theta^{\varepsilon} \cdot \int di_{\varepsilon i} \exp[-\xi - \beta \cdot h_{\varepsilon i}(i_{\varepsilon i}; W)] \ . \tag{4.10a}$$

For constant input signals, it is analogous to (4.5c and 4.6a-b):

$$H = \sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} P_{\varepsilon i} \cdot h_{\varepsilon i}(i_{\varepsilon i}; W) \ ,$$
$$\frac{\partial S}{\partial i_e} = -\beta \sum_{\varepsilon=1}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} P_{\varepsilon i} \cdot \left[\frac{w^2}{2} + \frac{(\Delta w)^2}{6} \right] \cdot A_{\varepsilon i, e} \ , \tag{4.11a}$$
$$\frac{\partial S}{\partial x_s} = -\beta \sum_{\varepsilon=1}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} P_{\varepsilon i} \cdot \left[\frac{W^2}{2} + \frac{(\Delta W)^2}{6} \right] \cdot \sum_{e \in E_{\varepsilon i}} \chi_{\varepsilon i}(e, s) \ .$$

In case of pulses as input signals we get:

$$f_{\varepsilon i}(a^0_{\varepsilon i}, \Delta^0_{\varepsilon i}; W) = \exp[-\xi - \beta \cdot h_{\varepsilon i}(\hat{i}_{\varepsilon i}; W)] \ , S = \xi + \beta \cdot H \ , \xi := \ln Z \ ,$$
$$Z = \sum_{\varepsilon=1}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} Z_{\varepsilon i} \ , \quad Z_{\varepsilon i} = t_d^{N-\varepsilon} \cdot \theta^{\varepsilon} \cdot \exp[-\xi - \beta \cdot h_{\varepsilon i}(\hat{i}_{\varepsilon i}; W)] \ . \tag{4.10b}$$

Similarly for (4.11a):

$$H = \sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} P_{\varepsilon i} \cdot h_{\varepsilon i}(\hat{i}_{\varepsilon i}; W) \ ,$$
$$\frac{\partial S}{\partial \hat{i}_e} = -\beta \sum_{\varepsilon=1}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} P_{\varepsilon i} \cdot A_{\varepsilon i, e} \cdot \left[\frac{w^2}{2} + \frac{(\Delta w)^2}{6} \right] \ , \tag{4.11b}$$
$$\frac{\partial S}{\partial x_s} = -\beta \sum_{\varepsilon=1}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} P_{\varepsilon i} \cdot \sum_{e \in E_{\varepsilon i}} \chi_{\varepsilon i}(e, s) \cdot \left[\frac{W^2}{2} + \frac{(\Delta W)^2}{6} \right] \ .$$

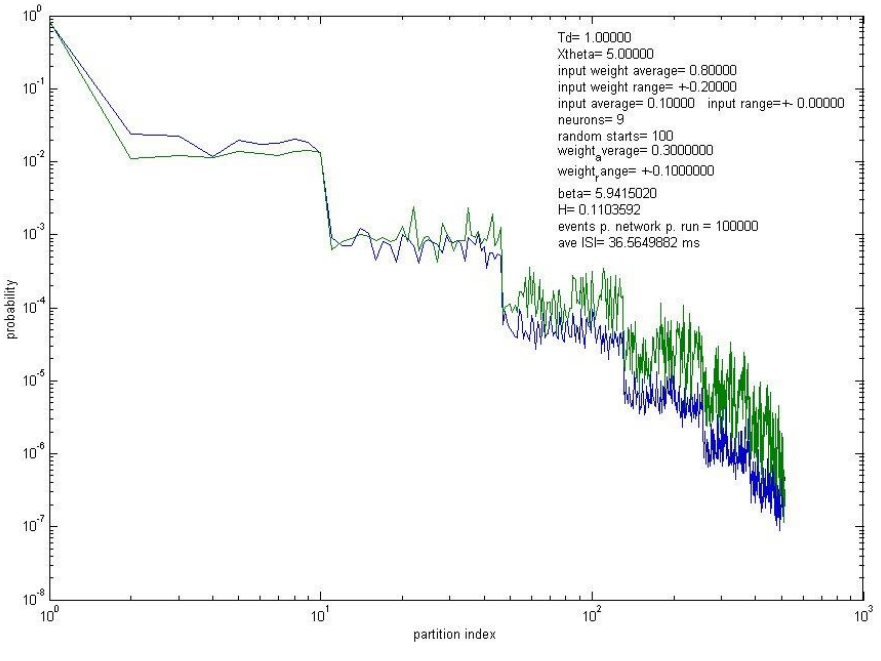
4.3 Agreement of Theory with Experiment

A. Synapses and Input Signals Known

Let us consider a fully connected net of 9 neurons. 100 thousand pulse events and 100 sets of initial values were spent per net. The input synapses had values of the interval $[0.6, 1.0]$, the other synapses of $[0.2, 0.4]$. The threshold was set to $\theta = 5$ and t_d to $1\,ms$ ($1\,ms$ was used as the unit, and accordingly synapses are measured in kHz). In the context of this section the term *distribution* means always the distribution of the frequencies of occurrence of the partitions, ie. the quantities $P_{\varepsilon i}$. In the following, these quantities are compared with the ones $P_{\varepsilon i} = Z_{\varepsilon i}/Z$ given by the theory.

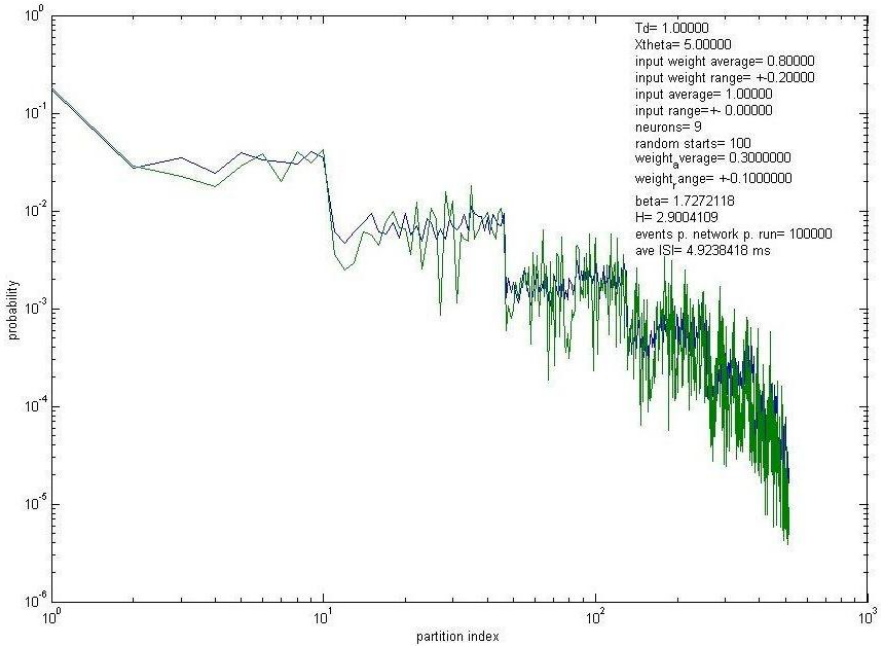
Representative of many simulations, Figures 4.1a, 4.1b show the distributions obtained with a constant input set to 0.1 and 1.0, respectively. In this chapter the experimental distributions are always painted *green*, the theoretical ones *blue*.

Curves decreasing from left to the right are characteristic for operating in the cold range. When increasing the input signal even further, the net predominantly transmits (Figures 4.1c). Despite a pulse rate of $673Hz$ the

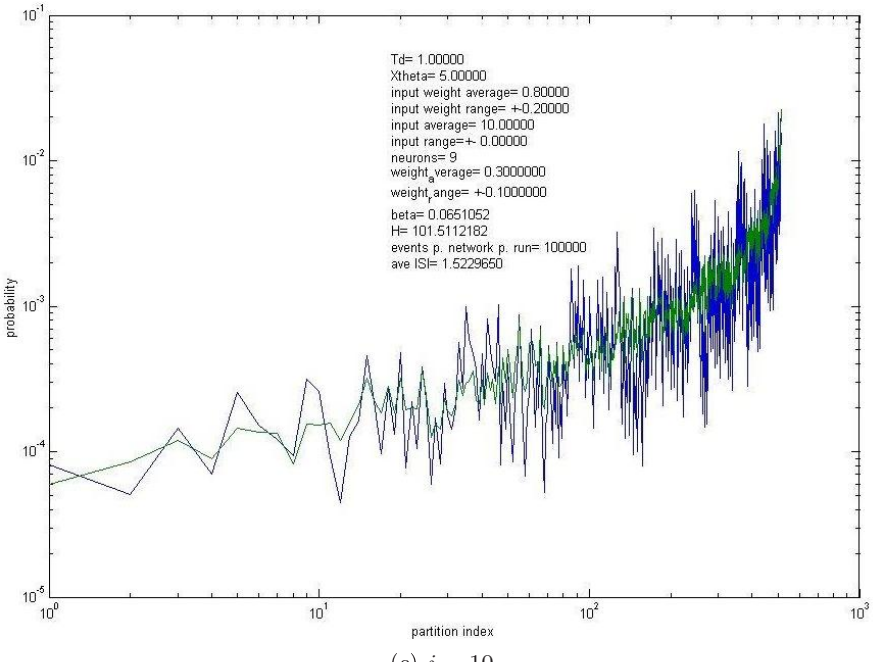


(a) $i = 0.1$

Fig. 4.1 Experimental and theoretical distribution with known synapses



(b) $i = 1$



(c) $i = 10$

Fig. 4.1 (continued)

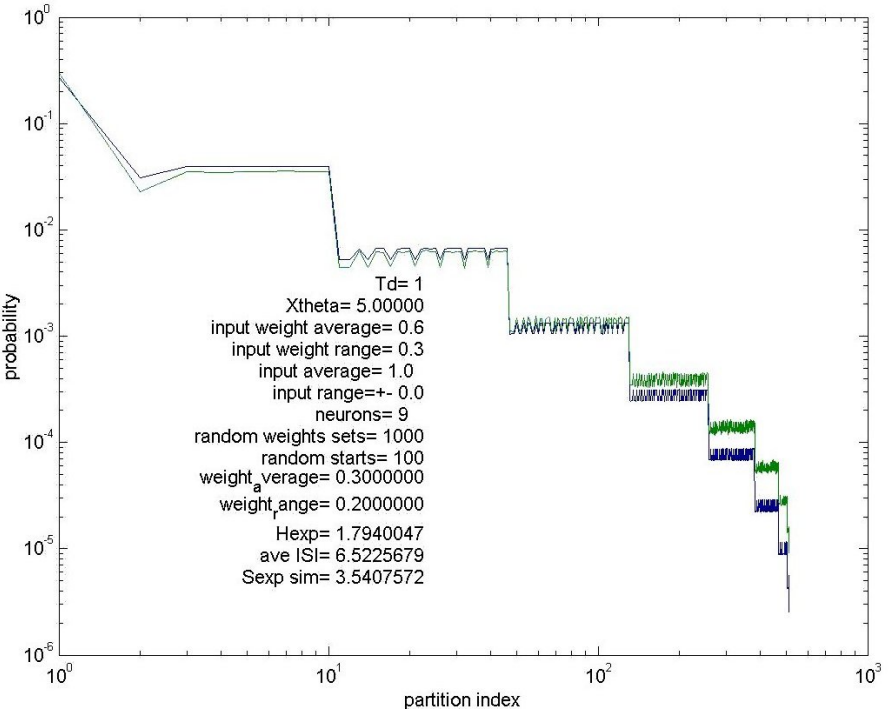


Fig. 4.2 Experimental and theoretical distribution averaged over 1000 nets with different, but representative synapses; $i = 1.0$

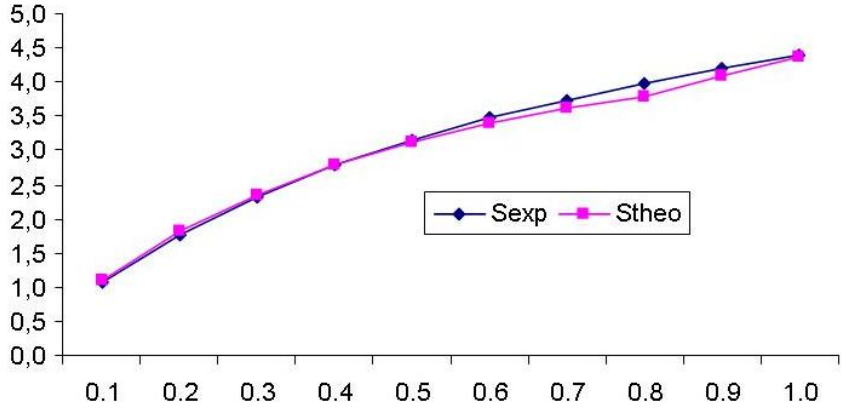
agreement between theory and experiment indicates the satisfactory modelling of the temporal and spatial means.

The performance of the modelling becomes particularly clear when 100 different sets of initial values and 1000 representative sets of synapses are considered, and the experimental as well as the theoretical distributions are averaged (Figures 4.2, after 11:32 hours of simulation).

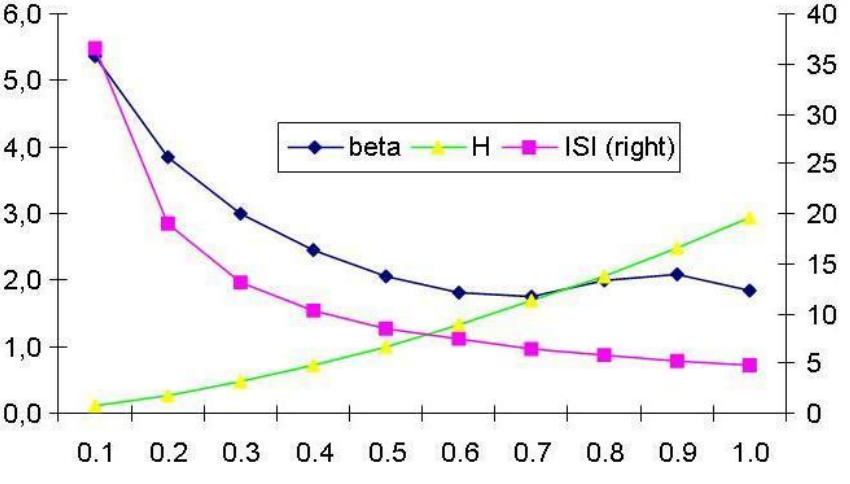
Because the entropy can be determined experimentally only by means of measuring the frequencies of occurrence of each partition as a function of the respective initial values for a_e and Δ_s, data types would have to be introduced with more than $128bit$ precision provided the intervals from 0 to t_d and θ, resp., are divided into 100 pieces/parts. To reduce the costs of programming and the time of simulation we do without the comparison of experimental and theoretical entropy. Instead, we use the function $-\sum P_{\varepsilon i} \cdot \ln P_{\varepsilon i}$ and the experimentally and theoretically, respectively, determined values of $P_{\varepsilon i}$. $-\sum P_{\varepsilon i} \cdot \ln P_{\varepsilon i}$ is designated as *partial entropy*. As the theoretical entropy contains besides this function still another function which is easy to compute (cf. Gl. (4.5d)), the comparison of these partial entropies is already sufficient for testing the agreement of theory and experiment.

Figure 4.3a displays the experimental and theoretical partial entropies, and Figure 4.3b shows H, the mean interspike interval (ISI) and β as the function of the input signal. With the input ranging from 0.1 to 1.0 the agreement is better than 3%.

In Figure 4.3c the experimental and theoretical pulse rates of the neurons are shown for an input signal of strength 0.8, and in Figure 4.3d the mean pulse rate (according to 3.24a) as function of the input signal. The suitability of the modelling is proven here, too.

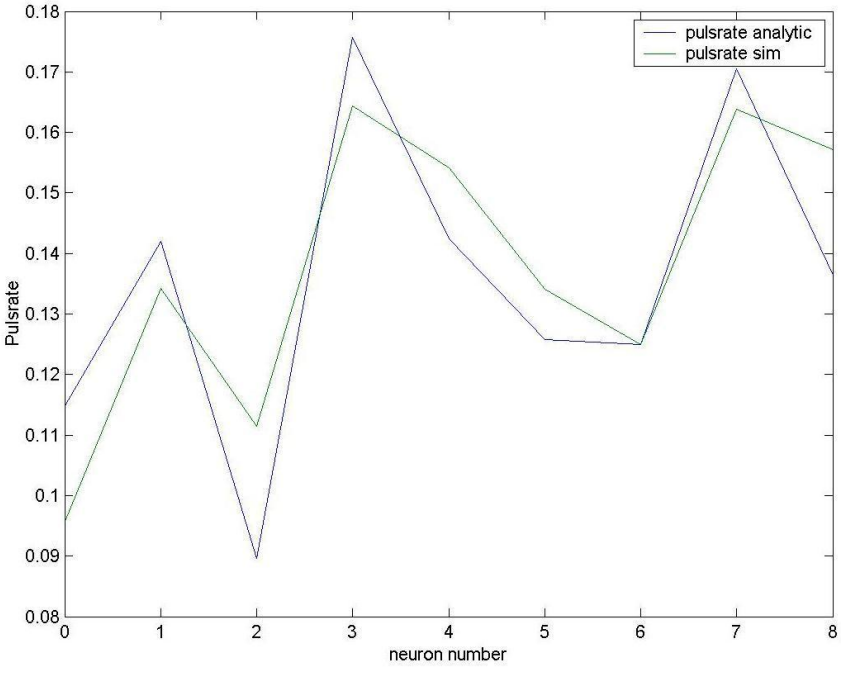


(a) Experimental and theoretical partial entropy as function of the input signal

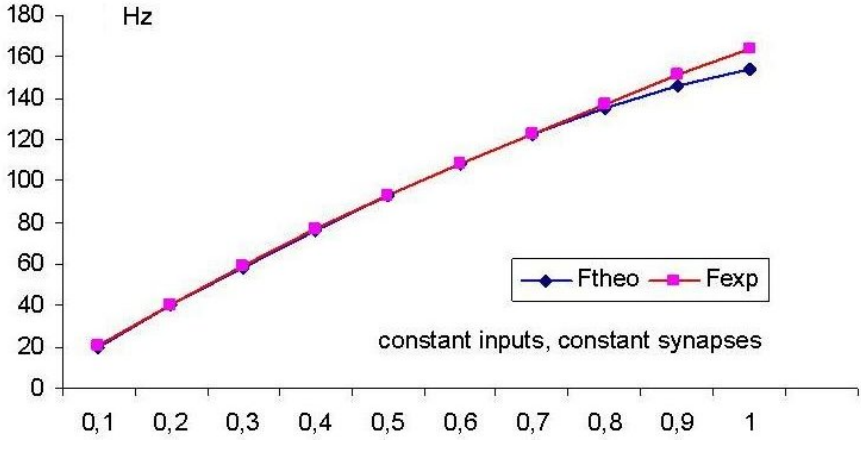


(b) Experimental energy H, β (left ordinate) as well as ISI (right ordinate) as function of the input signal

Fig. 4.3



(c) Experimental/theoretical pulse rates of the neurons at input 0.8



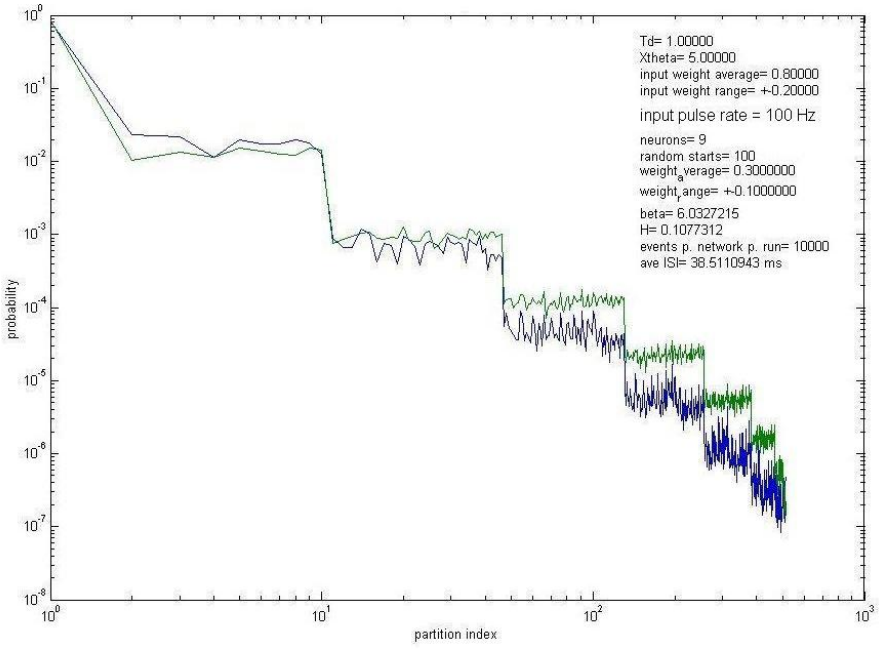
(d) Mean experimental/theoretical pulse rate as function of the input

Fig. 4.3 (continued)

B. Synapses and Distribution of Input Signals Known

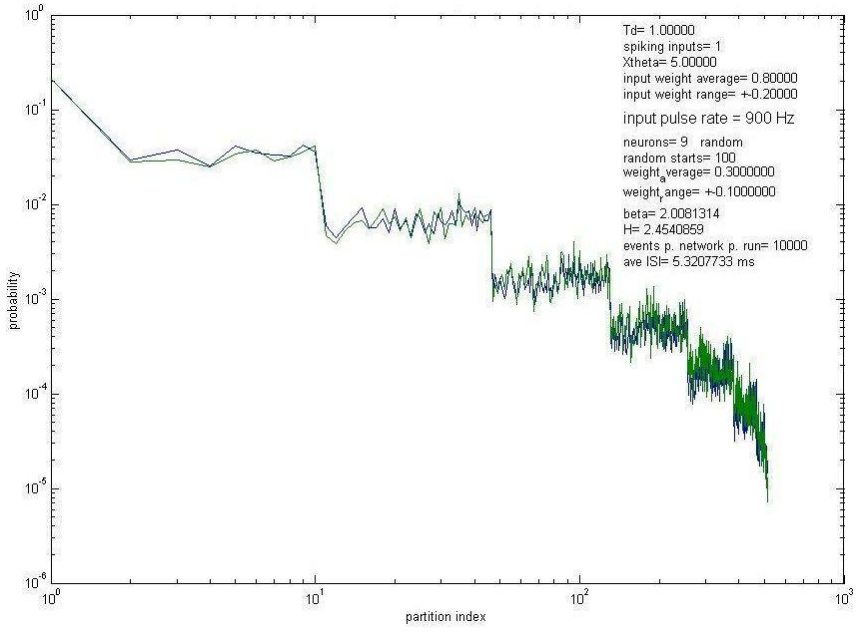
Instead of constant signals we now apply pulses to the input neurons. Pulses range between 100 and 900 Hz, thereby achieving comparability with the *constant input case* (100 Hz is equivalent to a constant signal of 0.1). As pulse generator a net of 9 neurons is used which is fed by constant input signals. The net is the same as the one from section A.

Figures 4.4a,b show the experimental/theoretical distribution for a fully connected net of 9 neurons at input pulse rates of 100 and 900 Hz, respectively. In Figure 4.4c the experimental/theoretical partial entropy is shown as a function of the mean input pulse rate. Figure 4.4d displays the curves for ISI, β and H. Finally, in Figure 4.4e we see the experimental/theoretical mean pulse rate plotted as a function of the input rate.

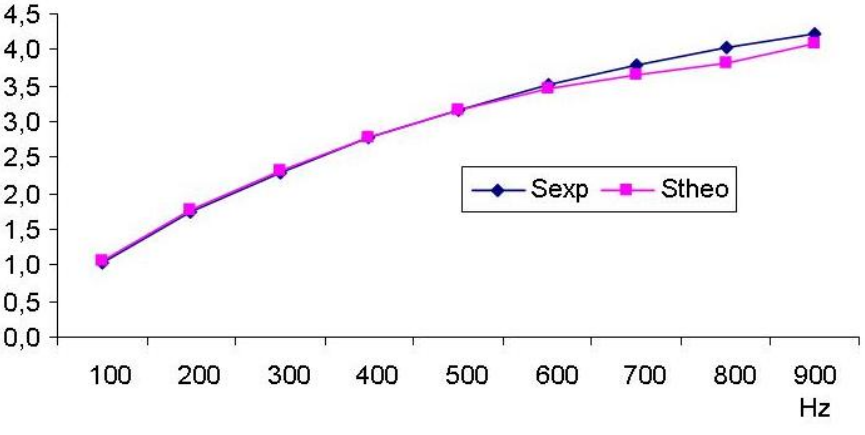


(a) Experimental/theoretical distribution for input pulse rate of 100Hz

Fig. 4.4

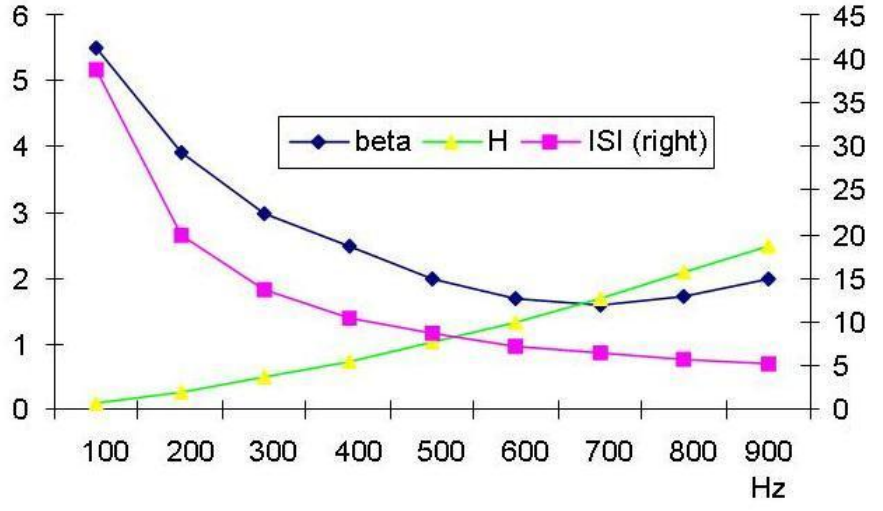


(b) dito, for input pulse rate of $900Hz$

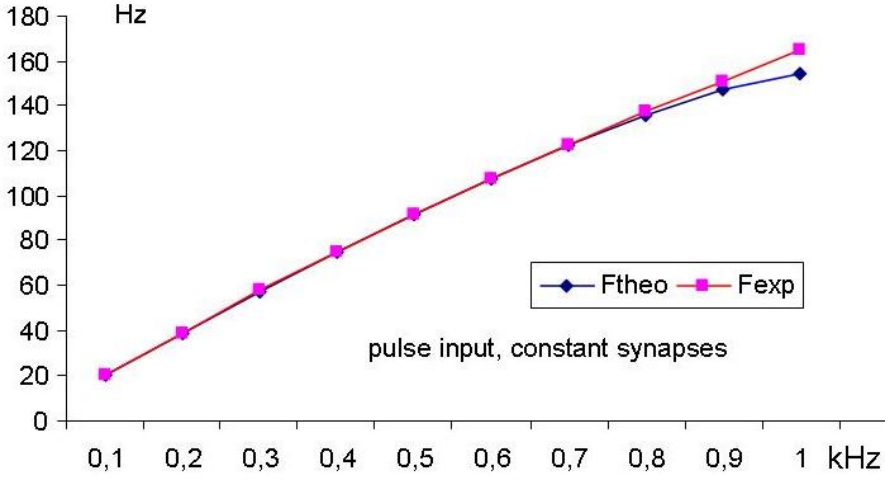


(c) Experimental/theoretical partial entropy as function of the input pulse rate

Fig. 4.4 (continued)



(d) Experimental energy H, β (left ordinate) and ISI (right ordinate) as function of the input pulse rate



(e) Experimental/theoretical mean pulse rate as function of the input pulse rate

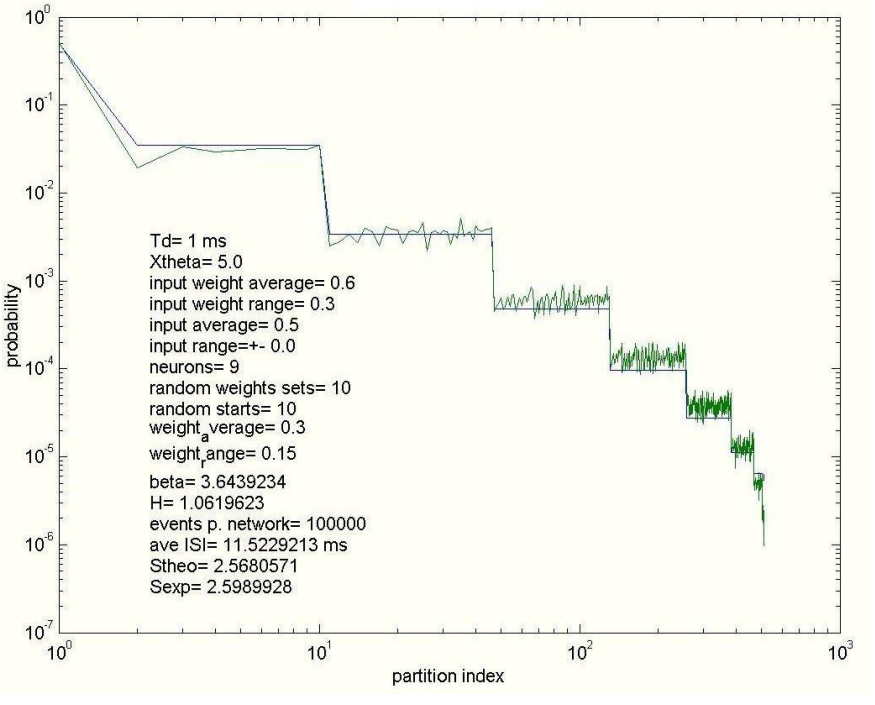
Fig. 4.4 (continued)

C. Input Signals and Distribution of Synapses Known

Again, a fully connected net of 9 neurons is considered, with 100 thousand pulse events per net and 100 different sets of initial values. The input synapses adopt values from $[0.3, 0.9]$, the other synapses values from $[0.15, 0.45]$, the distribution being equipartition (any other distribution can be used, provided it is known (cf. 4.9a)).

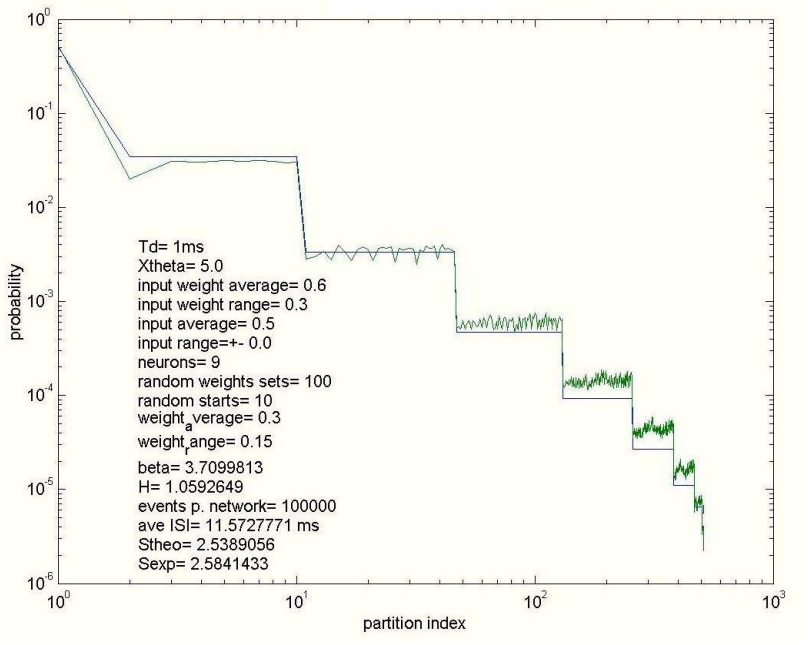
Figure 4.5a-c show experimental/theoretical distributions for $n = 10$, 100 and 1000 sets of synaptic representatives. Already with 100 sets of synapses a stationary distribution is obtained.

The experimental/theoretical entropy is displayed as function of the input signal in Figure 4.5d, the mean values of energy and ISI as well as the value of β in Figure 4.5e. Finally, the experimental/theoretical mean pulse rate is demonstrated in Figure 4.5f. Again, the modelling of the transformation of a temporal into a spatial mean seems to work satisfactorily.

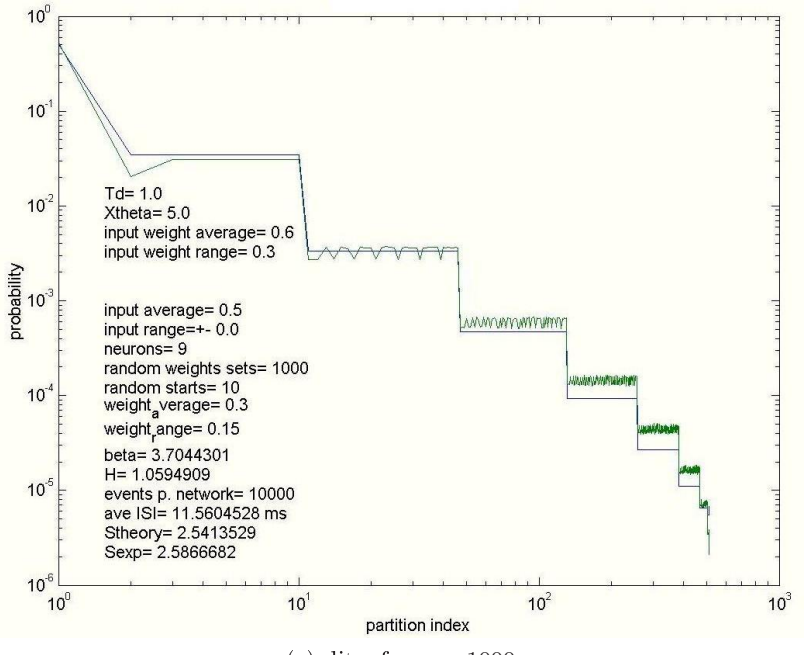


(a) Experimental/theoretical distribution for $n = 10$ synaptic sets and $i = 0.5$

Fig. 4.5

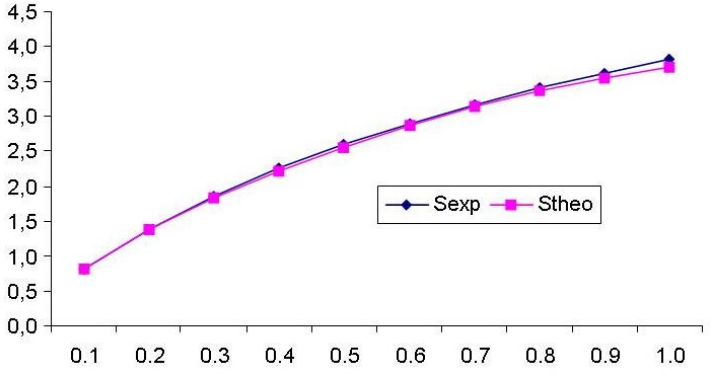


(b) dito, for $n = 100$

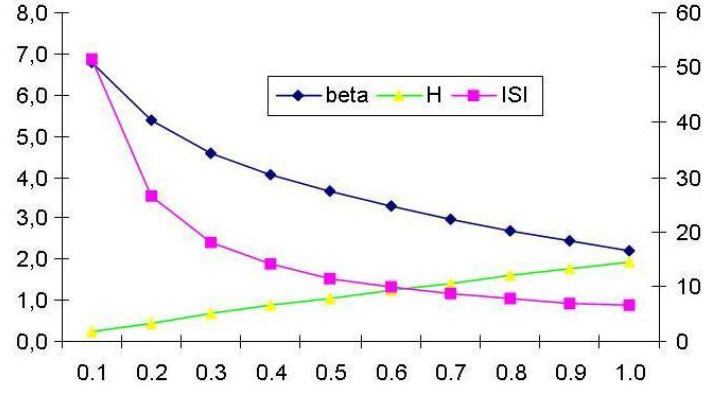


(c) dito, for $n = 1000$

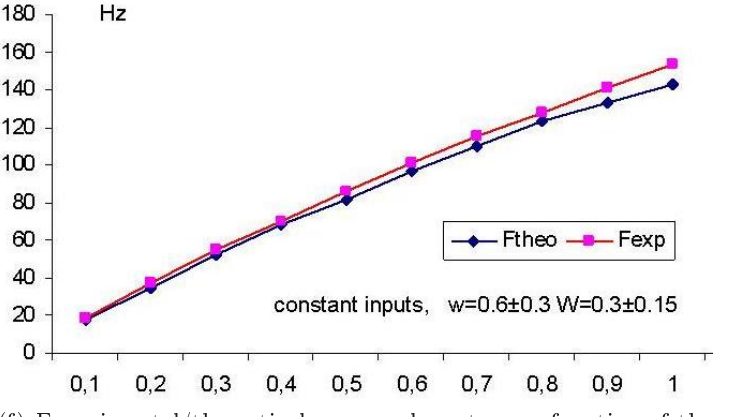
Fig. 4.5 (continued)



(d) Experimental/theoretical entropy as function of the input signal



(e) Experimental energy H, β (left ordinate) and ISI (right) as function of the input signal



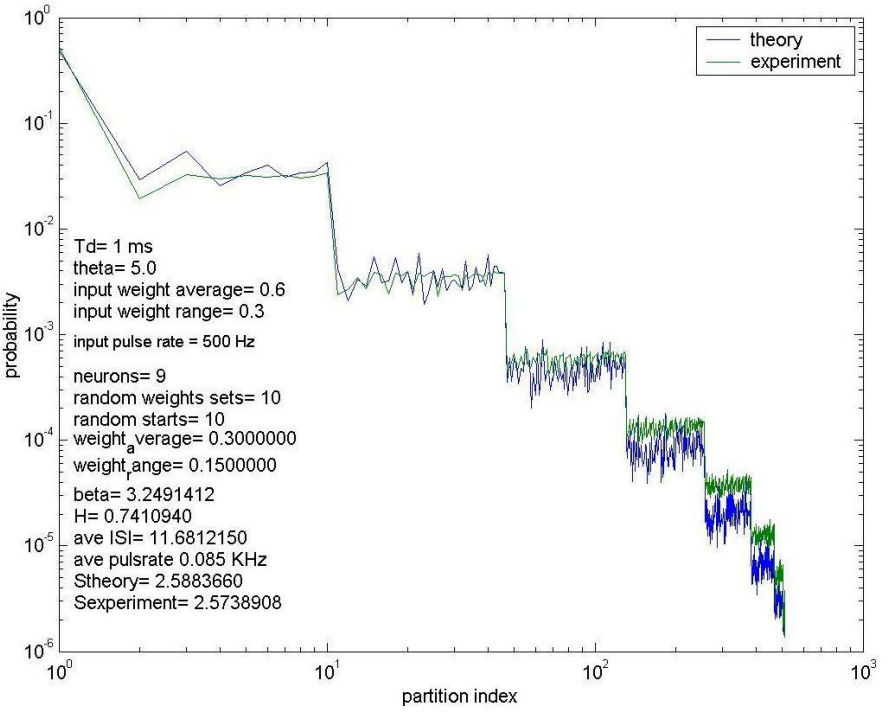
(f) Experimental/theoretical mean pulse rate as a function of the input signal

Fig. 4.5 (continued)

D. Distributions of Synapses and Input Signals Known

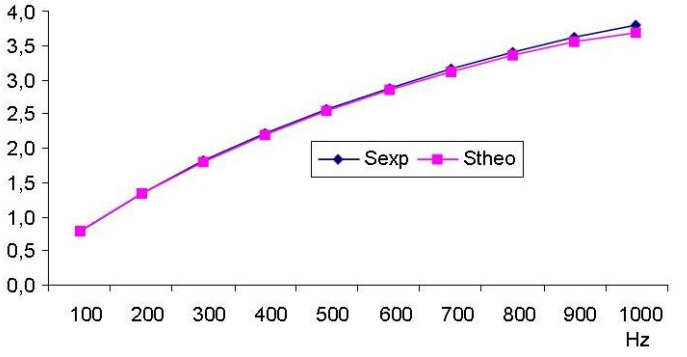
Instead of constant input signals we apply now pulses. The input synapses are chosen from $[0.3, 0.9]$ and the other synapses from $[0.15, 0.45]$.

Figure 4.6a shows the experimental/theoretical distribution, Figure 4.5b,c displays the curves for H, ISI and β as function of the mean input pulse rate. Lastly, Figure 4.6d demonstrates the experimental/theoretical mean pulse rate as a function of the input pulse rate[1].



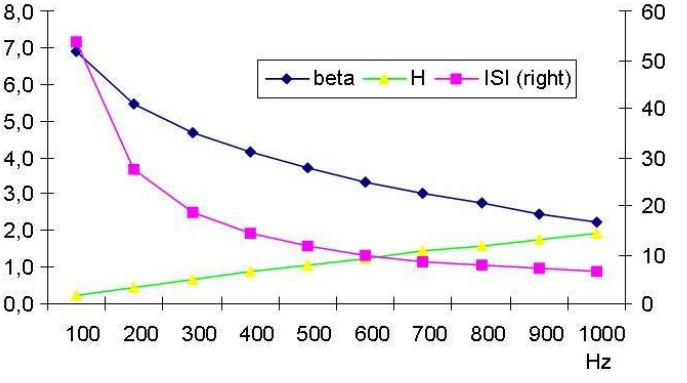
(a) Experimental/theoretical distribution for input pulse rate of $500\,Hz$

[1] The reader will have noted that threshold and pulse duration are used with values different from 1 in the above calculations. In chapter 2 we set both values to 1. Because the demand for precision in the simulation of the theory depends on the values for θ and t_d and a microelectronic realisation of the nets under consideration also can take advantage of the adjustability of these two values, we have used non-trivial values for threshold and pulse duration ever since chapter 3.

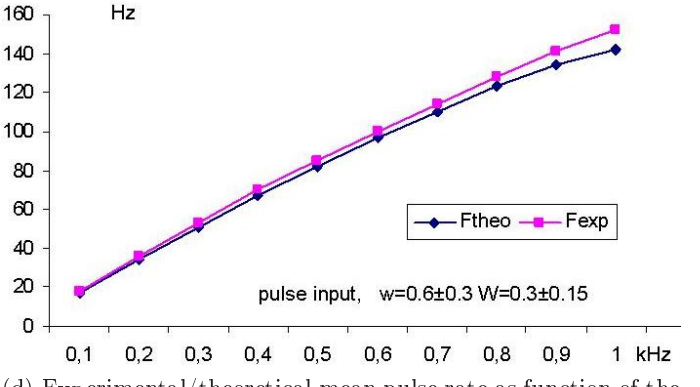


(b) Experimental/theoretical entropy as function of the mean input pulse rate

Fig. 4.5



(c) Experimental energy H, β (left) and ISI (right) as function of the mean input pulse rate



(d) Experimental/theoretical mean pulse rate as function of the mean input pulse rate

Fig. 4.6

4.4 Lack of Correlation

The absence of correlations between the firing times of any two neurons of a net operating in the cold range was observed in chapter 2. We now justify this fact by theory.

The expression for the entropy builds upon the logarithm of the volume (of the space) of all partitions and the product of β with the expectation value for the signal energy. The first term describes the size of the space of all partitions, the second term points to the approximate proportionality of entropy and energy. Since the signal equations and the signal energy depends only on the synapses and input signals, but not on membrane signals or other terms specifying the transition process from one partition to another one, there is no hint that not all partitions are allowed for the dynamics of the net. Therefore, a distribution was created by the maximization of the entropy which was carried by all partitions, featuring a single analytical structure for the distribution of all partitions. Thereby follows that every 2-neuron correlation is negligibly small or vanishes completely (in the limit of large nets). For let us assume a pair of neurons would exist with non-vanishing correlation. Then no partition could exist in which one of the two neurons in the pair participated as a transmitter, but the other one did not. Hence, the probability for the combination $(1e, 2s)$ or $(1s, 2e)$ would have to be zero. However, this is not the case as observation has shown. Consequently, there is agreement between theory and experiment also in this point.

4.5 Determining the Signal Energy and Entropy by Pulse Rates

Let us consider, for instance, the net for the creation of pulses controlling a muscle contraction, as described in chapter 2. It is a fully connected net of 100 neurons the entropy of which can not be computed directly.

Also, the mean energy H is hard to measure experimentally, let alone to be computed. For when such a net is implemented in silicon, from the power consumption of the net it is hard to derive the value for the signal energy as defined in (3.3).

In contrast, the measurement/simulation of pulse rate and mean pulse rates, resp., can easily be done and produced as function of the input signals. Therefore, the idea is to compute the quantity β from the relations (3.24a,b) for the pulse rate of the neurons of a net so that the value received for β in this way can be used for the determination of H in (4.5c, 4.8d), etc. For greater precision we start with the mean pulse rate. In the case of constant synapses and inputs we obtain the relation:

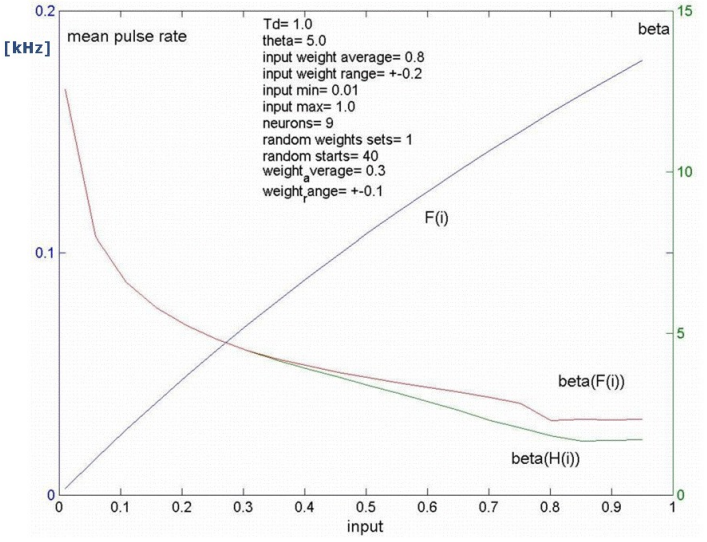


Fig. 4.7 Comparison of $\beta(F(i))$ with $\beta(H(i))$

$$F = \frac{1}{t_d \cdot N} \frac{\sum_{\varepsilon=1}^{N} (N-\varepsilon) \cdot t_d^{N-\varepsilon} \cdot \theta^{\varepsilon} \cdot \sum_{i=1}^{\binom{N}{\varepsilon}} e^{-\beta \cdot h_{\varepsilon i}(W_{\varepsilon i}, i_{\varepsilon i})}}{\sum_{\varepsilon=1}^{N} t_d^{N-\varepsilon} \cdot \theta^{\varepsilon} \cdot \sum_{i=1}^{\binom{N}{\varepsilon}} e^{-\beta \cdot h_{\varepsilon i}(W_{\varepsilon i}, i_{\varepsilon i})}} = \frac{1}{N} \cdot \frac{\partial \xi}{\partial t_d} \ . \quad (4.12a)$$

Hence, we get: $\beta = \beta(F(i))$. The comparison with the $\beta(H(i))$ curve results in a good agreement although only a single net with 10 different sets of initial values created the pulse rate. With more nets the β curves coincide even better. Note also the large input range which allows the pulse rate of the net vary from 0 to $170Hz$[2].

This way the explicit expressions for H and S can be computed:

$$H(i) = \sum_{\varepsilon=0}^{N} \sum_{i=1}^{\binom{N}{\varepsilon}} \frac{Z_{\varepsilon i}(\beta)}{Z(\beta)} \cdot h_{\varepsilon i}(i) \ , \quad S(i) = \ln Z(\beta) + \beta \cdot H(i) \ . \quad (4.12b)$$

In Figure 4.8 the theoretical H and S curves are displayed which were computed from $\beta(F, i)$. There is very good agreement between the respective experimental curves for H and S. Since for larger nets the variation of the pulse

[2] As mentioned already earlier, with larger nets the storage capacity of a PC is no longer sufficient. Then, it is recommended to set a bound for the probability of a partition (for instance, 10^{-6}). Those partitions which do not pass the bound are omitted from (4.12a). The value of β received this way is inserted into the reduced version of (4.5c), etc.

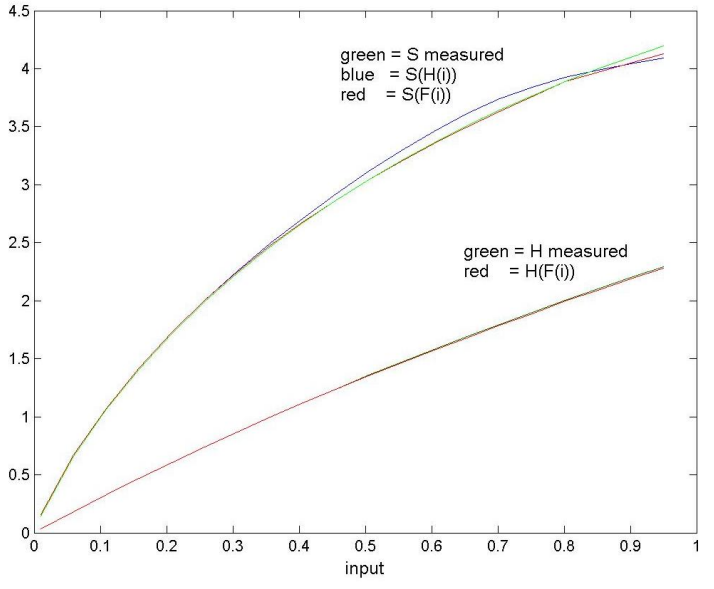


Fig. 4.8 Theoretical values for H and S as functions of $\beta(H,i)$ and $\beta(F,i)$, resp.

rates vanishes (cf. 3.24b and Figure 2.14(b)) a few neurons' pulse rates have to be measured simply to get the value for the mean pulse rate of the net.

In the appendix we present some instructions for using the simulator of the theory.

4.6 Summary

The theory is tested first with nets of constant synapses. The agreement between experimental and theoretical entropy is most striking. This holds for the cases of constant or pulsed inputs and for arbitrary distributions of synaptic values.

Just as the signal energy and its mean value are basic for the setup of the theory, so it is difficult to determine the signal energy values in an electronic or a biological implementation. Because of its ease of measurement, the pulse rate can be input into the theory instead of the mean of the signal energy and thereby signal energy and entropy recomputed. Again, the agreement of experimental and theoretical quantities is close to perfect.

Furthermore, nets have to fire pulses sparsely in order not to produce artificially particular pulse patterns.

5

Information Processing with Dynamic Synapses

Ulrich Ramacher and Arne Heittmann

As we had seen from the histograms for the *ISI* between two neurons (Figures 2.9 - 2.10, chapter 2) a net with constant synapses is unable to remain active in a particular subset of partitions only. Instead the net is carried by all partitions, this being the reason for the absence of patterns when operating in the cold range. Furthermore, it can recognise and differentiate, resp., spatial patterns only in a limited way because the permutation of input images can neither be separated by pulse rate nor entropy with restricted net size. Moreover, pulse patterns at the input side can never be mapped on to pulse patterns at the output side as long as the net operates coldly. On the other hand, *hot* nets are excluded because such nets hardly react to input signals.

This does not come as surprise, for nets with constant synapses in the cold operating range transfer entropy at the input side into an entropy of the net - each entropy arranged as a drumbeat of pulses which show no correlation. Consequently, only spatial-temporally averaged quantities such as pulse rates can be transferred as information. In contrast, should we want a net to reproduce local correlations (visible in the input image as connected sets of pixels of equal brightness or colour, for instance) we have to change the type of the net and choose one which permits the transfer, processing and recognition of information by entropy *and* correlation.

A new type of net is created by means of new types of neurons or synapses. Here the theory of the last chapter helps us to fathom whether changes at the neuron model will support the mapping of correlations. For the definition of a neuron's signal other than by a rectangular pulse or the inclusion of a membrane term indicating loss complicate the calculation of the state sums $Z_{\varepsilon i}$ and do not restrict the set of partitions - a necessary condition for the existence of correlations in a net. Thus, the synapses have to come up with some dynamic behaviour. In its local and causal form this behaviour can be controlled by the arrival of a pulse at a synapse and by the membrane potential. Equation 3.6 of chapter 2 expresses this concept by a simple differential equation.

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 83–104, 2010.
© Springer-Verlag Berlin Heidelberg 2010

How correlations will show up in the histograms and which elementary application steps are therefore made possible will be demonstrated in this chapter.

5.1 The Types of Solutions of Synaptic Equations

The equation 3.6 resulted from the synapse potential defined in (3.1) and is reproduced here for the reader's convenience:

$$\frac{dW_{es}(t)}{dt} = -\gamma \cdot W_{es} + \mu \cdot (a_e(t) - \theta/2) \ , \quad e \in E_{\varepsilon i}, s \in \{S_{\varepsilon i} \cap N_e\} \cup \{0\} \ ,$$
$$\frac{dW_{es}(t)}{dt} = -\gamma \cdot W_{es} \ , \qquad e \in E_{\varepsilon i}, s \in \{E_{\varepsilon i} \cap N_e\} \ .$$

It is assumed that synapses take on positive values only and are bounded. Renewed differentiation using (2.1a) results in:

$$e \in E_{\varepsilon i}, s \in S(t) := \{S_{\varepsilon i}(t) \cap N_e\} \cup \{0\} \ : \ddot{W}_{es} + \gamma \dot{W}_{es} - \mu \cdot W_{es} \cdot x_s = \mu z_{es} \ ,$$
$$e \in E_{\varepsilon i}, s \in \{E_{\varepsilon i}(t) \cap N_e\} : \ddot{W}_{es} = -\gamma \dot{W}_{es} \ ,$$
$$\text{with} \quad \dot{W}_{es} = \frac{dW_{es}}{dt} \ , \ \ddot{W}_{es} = \frac{d\dot{W}_{es}}{dt} \ , \ z_{es} := \sum_{s \neq s' \in S(t)} W_{es'} \cdot x_{s'}(t) + W_{e0} \cdot i_e \ . \tag{5.1}$$

Equation 5.1 possesses two classes of solutions depending on the sign of μ.

Case sign$(\mu) < 0$. The homogeneous solution reads (all $z_{es} = 0$):

$$W_{es}(t) = A \cdot \sin\left[\frac{\lambda}{2} \cdot (t - B)\right] \cdot e^{-0.5 \cdot \gamma t} \ , \quad e \in E_{\varepsilon i}, s \in S(t) \ ,$$
$$W_{es}(t) = C_0 \cdot e^{-\gamma t} \ , \quad e \in E_{\varepsilon i}, s \in \{E_{\varepsilon i}(t) \cap N_e\} \ ,$$

with λ being the square root of $4|\mu x| - \gamma^2 > 0$ and the constants A, B, C_0 derived from the initial values at time t_x (t_0 or $t_{\varepsilon i}$ of a partition):

$$A = \frac{W(t_x)}{\sin\left[\frac{\lambda}{2} \cdot \left(\frac{\lambda}{2} \cdot \arctan r\right)\right]} \cdot e^{0.5 \cdot \gamma t_x} \ , \ r := \frac{\lambda}{\gamma \cdot W(t_x) + 2\dot{W}(t_x)} \ ,$$
$$B = t_x - \frac{\lambda}{2} \cdot \arctan r \ , \ C_0 = W(t_x) \cdot e^{\gamma t_x} \ .$$

The structure of the inhomogeneous solution is revealed by:

$$W_{es}(t) = \frac{\lambda}{2} \int_{t_x}^{t} z_{es}(t') \cdot \sin\left[\frac{\lambda}{2} \cdot (t - t')\right] \cdot e^{0.5 \cdot \gamma \cdot (t' - t)} dt' \ . \tag{5.2a}$$

Thus, the synapse acts as a damped oscillator directed by the stimulus z_{es}. As we will soon see, such synapses do not support correlation. It is quite the contrary in the next case.

Case $sign(\mu) > 0$. The homogeneous solution now reads:

$$W_{es}(t) = C_1 \cdot e^{0.5\cdot(\lambda-\gamma)t} + C_2 \cdot e^{-0.5\cdot(\lambda+\gamma)t} \ , \quad e \in E_{\varepsilon i}, s \in S(t) \ ,$$
$$W_{es}(t) = C_0 \cdot e^{-\gamma t} \ , \quad e \in E_{\varepsilon i}, s \in \{E_{\varepsilon i}(t) \cap N_e\} \ ,$$

with λ the square root of $4\mu x + \gamma^2$. From the initial conditions the following constants result:

$$C_1 = \frac{(\lambda+\gamma)\cdot W(t_x) + 2\dot{W}(t_x)}{2\lambda} \cdot e^{-0.5\cdot(\lambda-\gamma)t_x} \ ,$$
$$C_2 = \frac{(\lambda-\gamma)\cdot W(t_x) - 2\dot{W}(t_x)}{2\lambda} \cdot e^{+0.5\cdot(\lambda+\gamma)t_x} \ ,$$
$$C_0 = W(t_x) \cdot e^{\gamma t_x} \ .$$

Again, the structure of the non-homogeneous solution becomes transparent in:

$$W_{es}(t) = \frac{\lambda}{2} \int_{t_x}^{t} z_{es}(t') \cdot \sinh\left[\frac{\lambda}{2}\cdot(t-t')\right] \cdot e^{0.5\cdot\gamma\cdot(t'-t)} dt' \ . \tag{5.2b}$$

Hence, the synapse is exponentially rising or decaying and depending on the stimulus z_{es}.

The next section reveals that synapses with positive μ are supportive of correlations and order, resp. Those with negative μ cause decorrelation and disorder, resp.

5.2 Synchronisation of Neurons

That the second case with positive μ deals with something novel can be explained by the 2-neuron system. It uses dynamic synapses, except for the input where constant and equal synapses apply. The input signal is equal for both neurons and ranging in [0,1], and without dynamic synapses, the pulse frequency of the neurons is set to $10\,Hz$, say. We further suppose that both neurons are receiving at the start time and the difference of the membrane potentials of the two neurons is less than half the threshold (Figure 5.1a). From (3.6) we conclude that both neurons tend to zero exponentially. Not before the membrane potential of one neuron (say, the second) has passed half the threshold - driven by the input signal only - and the other neuron (then, the first) has already reached the threshold and produced a pulse will the behaviour of the second synapse transit into the exponentially rising mode and the membrane potential of the second neuron shift closer to the threshold than could have happened without the first neuron's pulse, but just with the input signal. In consequence, the distance between the pulses of neuron 1 and 2 decreases.

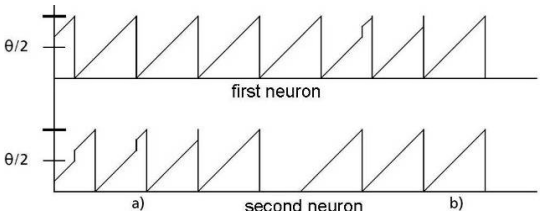


Fig. 5.1 Illustration of synchronisation in a 2-neuron system

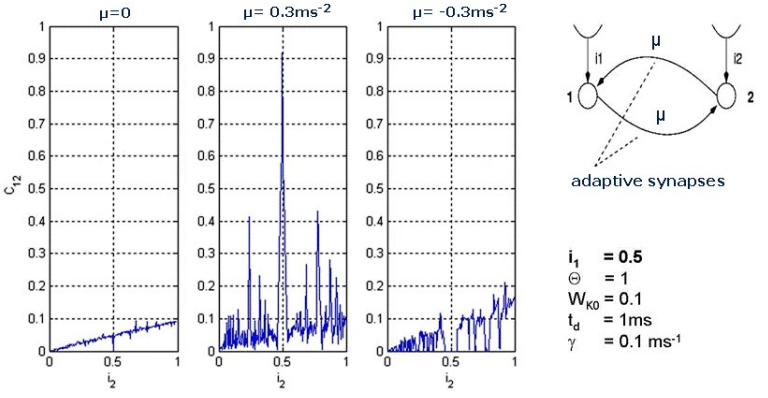
Consequently, the second neuron will fire before the first neuron has reached half the threshold. Thus, the first neuron is only driven by its input. When the first neuron fires a second time the second neuron will approach the threshold even more closely than at the first arrival of the first pulse, ie. the pulse difference between the two neurons further decreases. Because of the finite pulse duration the pulse difference reduces to about a fraction of $1\,ms$, but never reaches zero (not visible in the figure). Figure 5.1b shows that the first neuron has the active part of the synchronisation provided the distance of the membrane potentials at start time is larger than half the threshold.

Anyway, when the synchronisation of the neurons is damaged (for instance, by unequal input signals or noise) the synapses try to restore synchronisation. In other words: the dynamic synapses bring about a shortening or a delay in the accumulation process of a neuron, and aim at zero difference between pulses of the two neurons. Finally, the two neurons will fire almost simultaneously. This is called synchronisation.

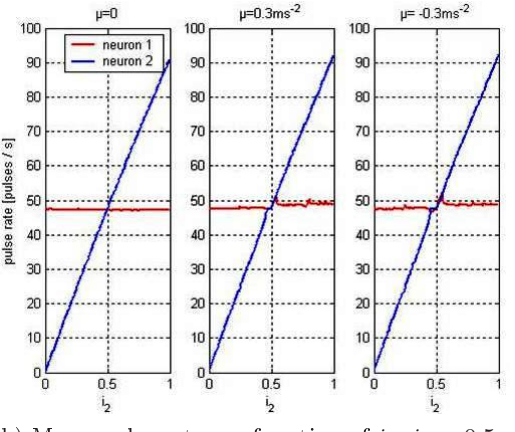
A similar behaviour can not be obtained with negative μ as is shown by Figure 5.2a. It displays the correlation of the pulse activity of the 2 neuron system for vanishing, positive, and negative μ. Here the input synapses are set to $0.1, \mu$ to $0.3\,ms^{-2}, \gamma$ to $0.1\,ms^{-1}, T$ to $20\,s$ and i_1 to 0.5, and the following definition of correlation is used:

$$C_{kl} = \frac{\int \chi(x_k)\,\chi(x_l)}{\int \chi(x_k)\,\chi(x_k)}\ , \quad \chi(x_n) = \begin{cases} 1 & \text{if neuron } n \text{ transmits} \\ 0 & \text{otherwise .} \end{cases}$$

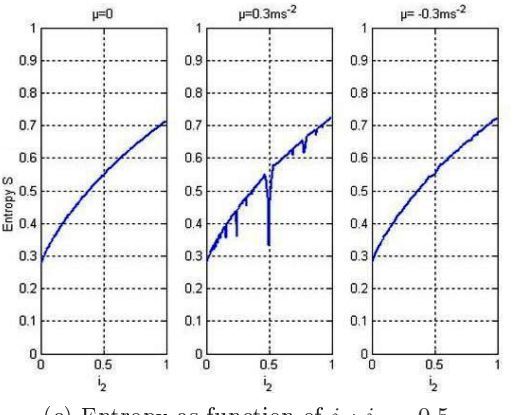
The nature of synchronisation to create a *collectively controlled isolation* of the neurons is demonstrated in Figures 5.2b,c. The pulse rate is practically independent of μ. Only the entropy breaks through when input signals become equal. This is easily understood: synchronisation allows only the partitions of all neurons to be either transmitters or receivers, ie. reduces the number of partitions by half. Thus we have arrived at a pattern generating process in a simple, but surprising manner: synchronisation of the transmission of pulses at almost equal times. This pattern generating process can be attached uniquely to the input image which consists of 2 pixels of equal brightness: the spatial-temporal correlation is mapped into the spatial-temporal correlation



(a) Correlation in a 2 neuron net as function of i_2; $i_1 = 0.5$



(b) Mean pulse rates as function of i_2; $i_1 = 0.5$.



(c) Entropy as function of i_2; $i_1 = 0.5$.

Fig. 5.2

of the simultaneous pulse activity of two neurons. Note that the pulse rate is proportionate to the brightness of the pixels.

We repeated the above experiment on a larger scale with 14 neurons (Figure 5.3). If all input signals are of equal value, a fully connected net will respond with almost simultaneous firing of pulses: after a short settling-in period and within $0.2\,ms$ after the first neuron started firing, almost all neurons started to fire, and within $0.2\,ms$ after the first neuron finished firing, almost all neurons had finished firing too. In an ideal synchronisation, all neurons in the net would behave as if they had started with the same initial values and no connecting synapses between neurons would be present. In contrast, a negative μ results in a pulse activity, as if constant synapses were implemented: the correlation at the input side is not copied by the dynamics of the net, but destroyed.

Basically the same, but of somewhat different appearance is the next-neighbouring connected net of 20×24 neurons which is exposed to an image of 20×24 pixels as in Figure 5.4a. Here the completed synchronisation takes longer. At first, depending on the initial values, 2-neuron subsystems will form at various locations which try to synchronise and bring about the master-slave behaviour described above. The various pairs of neurons compete and disturb each other. Those pairs with favourable time differences will succeed and synchronise the master neurons of adjacent 2-neuron subsystems, thereby growing to larger synchronised subsystems which produce small pulse waves. Whenever two pulse waves meet each other they unite and produce a larger pulse

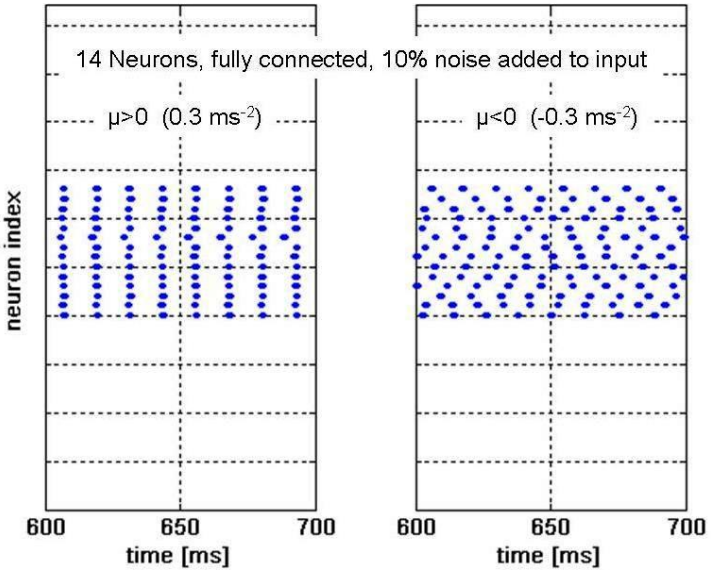
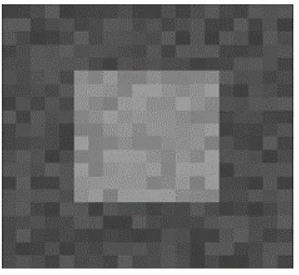
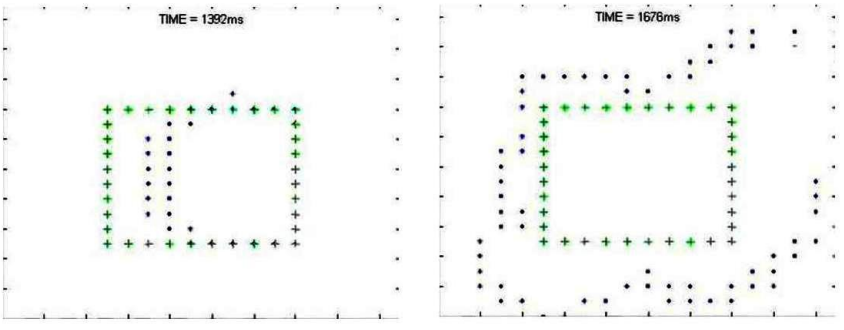


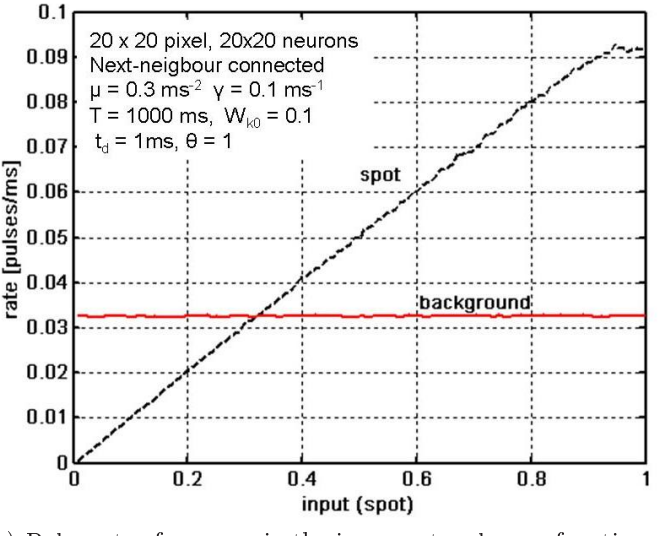
Fig. 5.3 Pulse activity of 14 neurons with $\mu = 0.3$ (left) and $\mu = -0.3$ (right)



(a) Test image with light rectangle on dark background and 10 % noise



(b) Wave fronts - settled and synchronised



(c) Pulse rate of neurons in the inner rectangle as a function of brightness

Fig. 5.4 Synchronisation

wave. Finally, several pulse wave fronts, or even a single one, move over the net. The velocity of a pulse wave is proportionate to the brightness of the pixels it is produced by. It is recommended that others perform some experiments themselves and look at the activity figures of the net in order to obtain some insight into the plethora of wave forms produced by synchronisation in neighbouring connected nets.

Figure 5.4b shows an extract of the pulse activity with two spreading waves: one runs in the inner rectangle at a higher velocity (left) than the other running in the background (right). To be noted is that both wave fronts can only move in their domains and do stop at the boundary of the rectangle. This indicates that synchronisation can bring about segmentation (more in the next section).

Obviously, a neuron directly situated in front of the wave front is on the verge of firing and reaches the threshold in a fraction of $1ms$, the energy for which is provided by its exponentially rising synapse, thereby feeding the growing wave. Since synapses decay exponentially fast, too, the neurons at the back of the wave front are only stimulated by the input signals. Hence, only those synapses in the net are active which are situated directly in front of the wave front, and they bring about the synchronisation with minimal energy.

The dependency of the mean pulse rate of the neurons in the inner rectangle on the input signal and pixel brightness, resp., is shown in Figure 5.4c.

Finally in Figure 5.5 an extract of the pulse activity in the same net with the same image is displayed when μ is negative. No pattern is emerging, and no

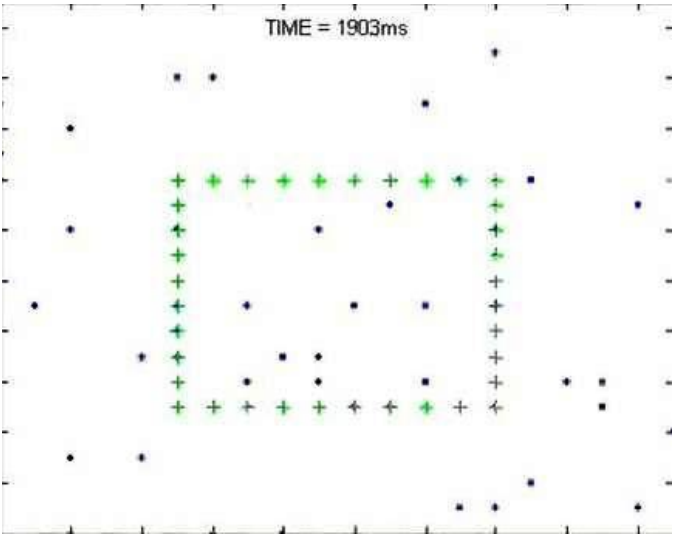


Fig. 5.5 Disorder produced with negative μ for image from Figure 5.4a

sign of order. It looks like oscillatory synapses can not produce any collective „sense“[1].

Synapses with positive μ can be regarded as a species of the class of excitatory synapses, and those with negative μ as representatives of the class of inhibitive synapses. In the literature, the usage of inhibitive synapses is connected with negative synapses (ie. the value is negative) whereas our synapses always take on positive values. Our proposal is based upon the fact that synapses with negative μ destroy order and synapses with positive μ preserve order.

5.3 Segmentation per Synchronisation

First of all we test the ability to segment with a virtual scene. In the left part of Figure 5.6 there is the image, and in the right part the bordered sets indicate the region of synchronisation and pixels of equal brightness, respectively. The bordered sets form silhouettes of the items in the picture, with one item normally consisting of several such sets.

In a natural scene the formation of silhouettes remains steady and it stays firm when there are variations of illumination (Figure 5.7b).

If the resolution of the image is quadrupled and now 65888 neurons with 262520 synapses are employed, the silhouettes seem to become richer and sharper (Figure 5.7c). It becomes apparent that the transitions from one region of synchronisation to the other are not represented. This is no wonder when using only a „spot“ detector, for this is what the neighbouring-connected net with positive μ forms indeed. In the next section we will deal with nets which produce pulse differences, and with these sets of constant gradient can be shown.

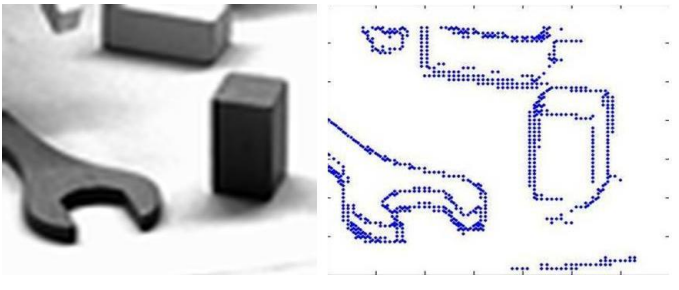
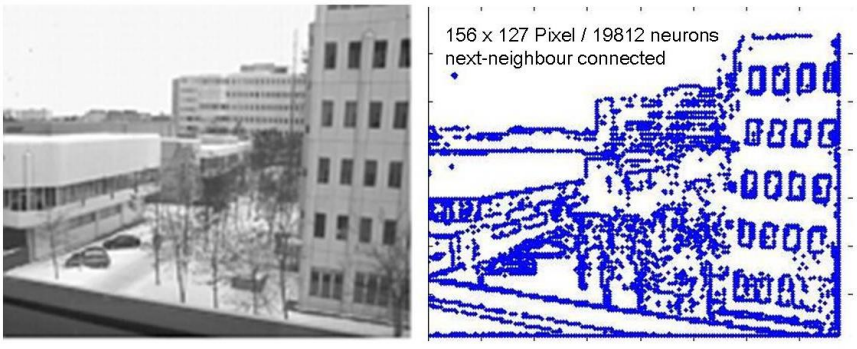
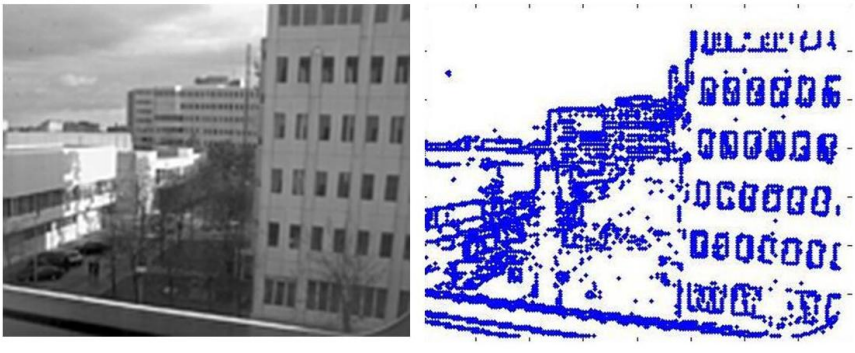


Fig. 5.6 Segmentation of a virtual scene (70×70 pixel and neurons, next-neighbouring connected)

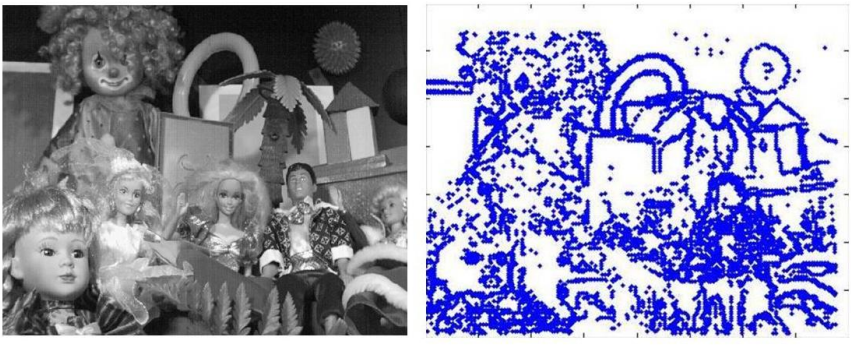
[1] In case of $\mu < 0$ the equations (5.1) represent a system of coupled damped oscillators. Such systems are of elementary importance in Physics. That these systems can not produce synchronisation, but short-lived beats are well-known. The case $\mu > 0$ has not yet been considered in Physics; emergent phenomena may profit from the positive μ case.



(a) Segmentation of a real image of 156×127 pixels



(b) dito, with different illumination



(c) Segmentation with resolution quadrupled (284×232 pixels)

Fig. 5.7

5.4 Calculation of Pulse Differences and Sums

Beside the nets with purely excitatory synapses ($\mu > 0$) those are of importance which mix both types of synapses. They enable us to compute pulse differences and sums.

Let us consider two neurons which are connected each to a pixel via a synapse. The neurons are not interconnected, just change pixel current into pulses. The input synapses are of equal and constant value. If the pixels are locally adjacent it makes sense to realise the pixel gradient as pulse difference. For this purpose the net outlined in Figure 5.8 is used.

The lines from neurons N_1, N_2 to neuron N_3 typify the dynamic synapses. Besides there is a further neuron N_4 the dendrite of which is connected to N_3. A dynamic synapse controlled by N_1 as well as N_3 is seated on this dendrite. As will be seen in a moment, the differential equation of this synapse on the dendrite side features the same structure as the synapse on the membrane side - the first one has two pulses, the other one a pulse and a membrane signal which drive the synaptic behaviour.

The neuron N_3 has no external input, thus always starts from *zero* membrane potential. Without a signal from N_1 and because of the negative μ the membrane potential of N_3 will increase rapidly according to (3.6) as soon as a pulse of N_2 is available. Not before the membrane potential has passed half the membrane potential will further growing be damped, strongly depending on γ_2 and μ_2. This way N_3 will be preloaded by N_2 to a value above half the

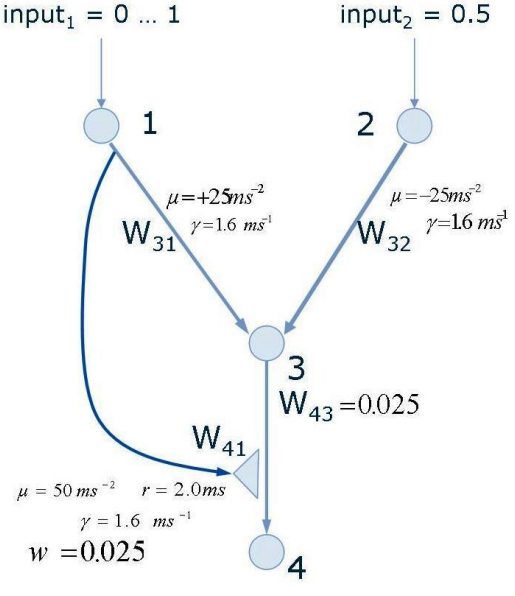


Fig. 5.8 Net computing the difference of two pulse streams

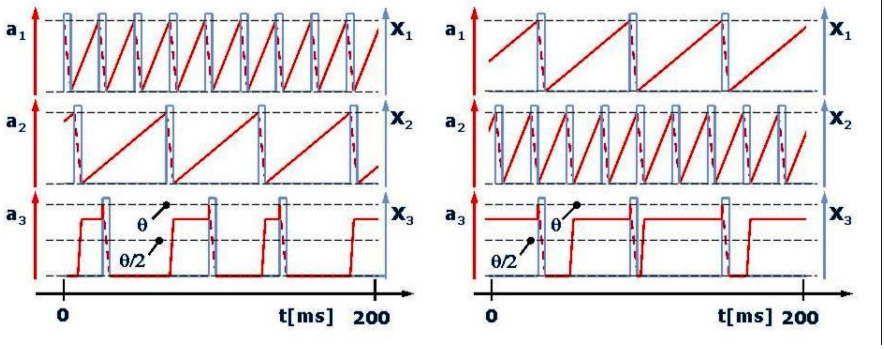


Fig. 5.9 Preloading and firing of neuron N_3

threshold, but never reach the threshold without a pulse from N_1. In particular, further pulses from N_2 will not contribute any more because the synapse acts as a block.

Because of the positive μ_1 and the membrane potential $> 0.5 \cdot \theta$ a pulse from N_1 causes N_3 to fire. Choosing coefficients suitably, a single pulse from N_2 suffices to preload N_3, and a single pulse of N_1 effects the firing of N_3 (Figure 5.9).

Conversely, when N_3 is not preloaded, a pulse of N_1 will never allow N_3 to fire (3.6). Thus, the simple net exemplifies exactly the situation that N_2 fires a pulse followed by N_1 firing pulses. This represents a „2 before 1“ temporal relation which is implemented by this mixed set of dynamic synapses.

How many pulses will have been received by N_1 before N_2 receives a second pulse has not yet been implemented so far. To compute the difference we need neuron N_4. This second part of the net uses a dynamic synapse W_{41} which is seated on the dendrite. The membrane potential of N_4 is given by the following equation:

$$a_4(t) := \int_{t_0}^{t} i_D(t')dt' \ , i_D(t') = W_{41}(t) \cdot x_1(t) + W_{43} \cdot x_3(t) \ , \tag{5.3a}$$

and the differential equation for the dendritic synapse reads:

$$\frac{dW_{41}}{dt} = -\gamma \ \cdot \ (W_{41} - w) + \mu \cdot \left(\int W_{41}(t') \cdot x_1(t')dt' - r \cdot x_3(t) \right) \ , \ \mu, r > 0 \ . \tag{5.3b}$$

w is the quiescent value of the synapse and r is chosen such that $\int W_{41}(t') \cdot x_1(t')dt' - r \cdot x_3(t) < 0$ holds if N_3 transmits a pulse. When N_3 produces a pulse (almost simultaneous with a pulse from N_1) W_{41} will become zero immediately, and N_4 accumulates a very small current only. When N_3 does not transmit, W_{41} grows exponentially with a pulse from N_1. With suitable choices of the coefficients μ, r, almost every pulse of N_1 will cause N_4 to

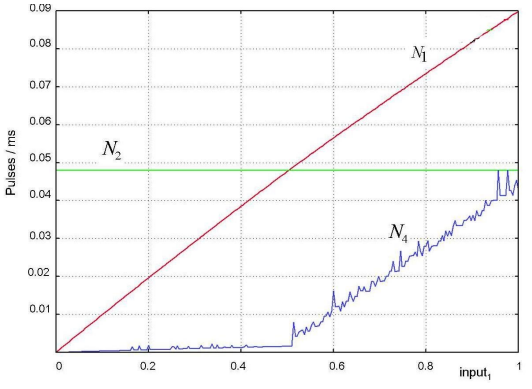


Fig. 5.10 Pulse rates of N_1, N_4

fire. Therefore, on average N_4 will produce the pulse difference between N_1 and N_2.

In Figure 5.10 the pulse rates of N_4 and N_1 are displayed as a function of the input signal. N_2 is presented as constant input.

The determination of the pulse difference works as long as the pulse rate of N_1 is larger than the one of N_2. If this is not the case, N_4 repeats the pulse rate of N_1. A symmetrical arrangement is delineated in Figure 5.11. Here neuron N_{4a} responds when $i_1 > i_2$, otherwise N_{4b} does[2].

The above arrangement can be generalized to cope with the difference of the sum of pulses produced by the additional inputs $N_{11} \ldots N_{1n}$ and the pulses of N_2. For the formation of pulse sums the connection of synapses to neuron N_3 can be abandoned, only those connecting to N_4 have to be kept. Note that the dendrite branches to all input neurons N_{1n}. Inspection of (5.3b) teaches that the dendritic synapses can be replaced in this special case

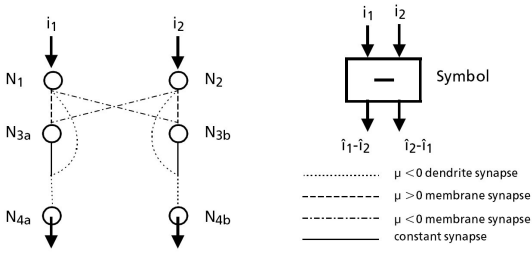


Fig. 5.11 Net for symmetric computation of pulse differences

[2] Because of its fundamental importance, it must be highlighted that Arne Heittmann found the configuration Figure 5.8 of neurons and synapses for the purpose of identification of a pulse difference.

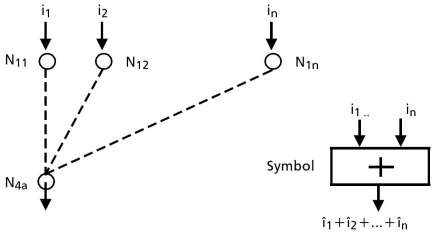


Fig. 5.12 Net for calculation of pulse sum

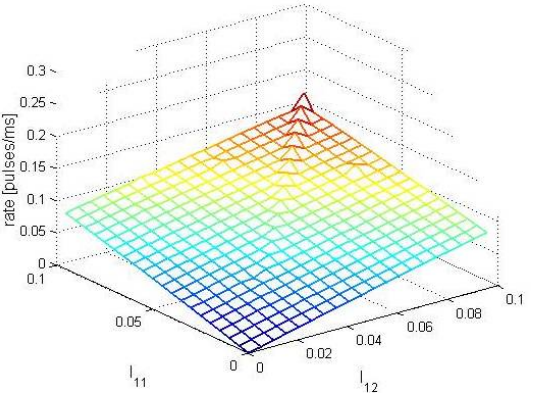


Fig. 5.13 Sum of pulses produced by the net of Figure 5.12

by simple dynamic synapses with threshold set to zero (Figure 5.12). Then, quite a linear process results (Figure 5.13). Clearly, the sum formation can not be exact in the mathematical sense since pulses of $N_{11} \ldots N_{1n}$ can arrive simultaneously. Thus, there is always a small loss (Figure 5.13).

5.5 Simple Applications

Let us consider two nets operating in the cold range and design a third net which gives the pulse sum or difference of the first two nets. It suffices to use constant synapses for the two input nets - only the third net uses dynamic synapses.

Arrangement for Pulse Summation. The two nets have 11 fully connected neurons, and 4 neurons are randomly chosen from each net each of which is connected by a constant synapse (equal for all 4) with the 4 sum nets (Figure 5.14).

The outputs of the sum net are connected to a single neuron with equal and constant synapses, the value of which is chosen such that the pulse rate

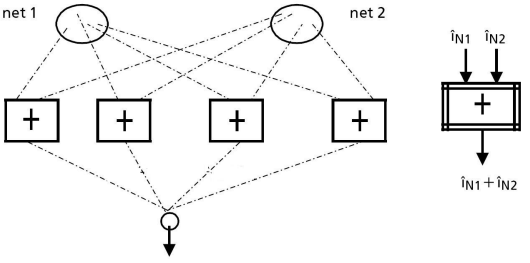


Fig. 5.14 Net for summation of the pulses of net 1 and 2

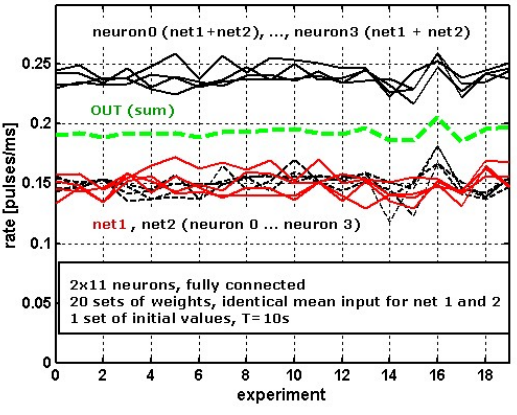


Fig. 5.15 Pulse rates of net 1 (black) and 2 (red). The summation (upper black lines) is affine linear with offset 0.53. Final output mid (red).

of the last neuron is proportionate to the averaged sum of the 4 input pulse rates (Figure 5.15).

Arrangement for Pulse Differences. As above. Besides, all first and second outputs of the difference nets are connected with a left and right neuron, resp., by means of equal and constant synapses (Figure 5.16 left).

The value of the constant synapses is dimensioned such that the pulse rate of the left and right neuron, resp., is an affine linear function of the mean of the 4 pulse difference rates. When all pulse rates of the net 1 are lower than the ones of net 2, the right output neuron will fire with the pulse difference rate of the two nets whereas the left neuron will pulse minimally (Figure 5.16 right).

Let us now suppose not all of the pulse rates of net 1 are higher than the ones of net 2. Then both output neurons add to the pulse activity. This is the case with identical nets and identical inputs in particular, when both output neurons present the same average pulse rate. As we would expect a

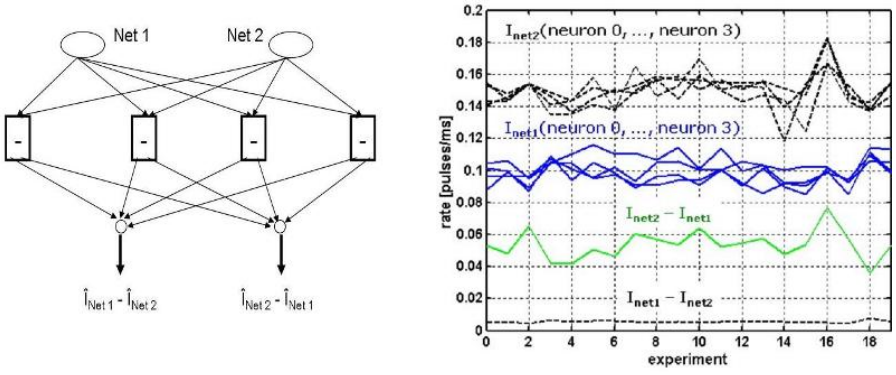


Fig. 5.16 Net for pulse differences (left), pulse differences (right)

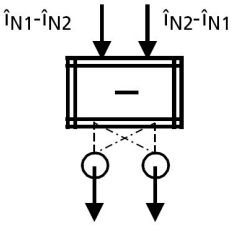


Fig. 5.17 Extension for pulse differences

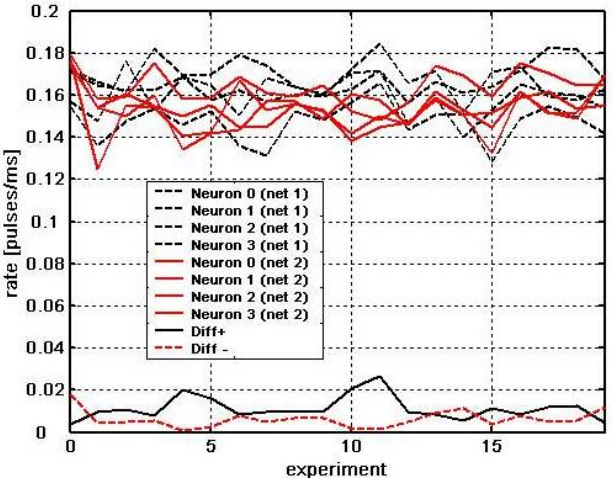


Fig. 5.18 Pulse rate difference of net 1 and 2 in Figure 5.17

zero pulse difference rate, it makes sense to add a symmetric difference stage (Figure 5.17). This results in a good convergence to zero (Figure 5.18).

Synchronisation Detector. The dendritic synapse is a function of two signals. Its differential equation reads:

$$\frac{dW_{es}}{dt} = -\gamma(W_{es} - w)(t) + \mu \cdot \left(\int W_{es}(t') \cdot x_s(t') dt' - r \cdot i_s(t) \right) . \quad (5.3c)$$

Here w is the quiescent synaptic value and i_s the dendritic current acting on the synapse which may represent a single pulse (as in (5.3b)) or a weighted overlaying of several pulses. For $\mu > 0$ the synapse behaves (see above) as an asymmetric anti-coincidence detector, and for $\mu < 0$ it acts as a coincidence detector if the easily fulfilled condition $\int W_{es}(t') \cdot x_s(t') dt' - r \cdot i_s(t) < 0$ is in play. For with either x_s or i_s active, the synapse transmits only a very small amount of current into the dendrite. If both are active at the same time, W grows very rapidly - and thereby the current $W_{es} \cdot x_s(t)$ delivered by the synapse - before the synapse returns to its quiescent value.

A multitude of functions can be generated from this basic behaviour for the neuron when a sequence of positive and negative μ synapses settle at the dendrite.

Of particular importance is the case with synapses of exclusively negative μ (Figure 5.19). The total current delivered to neuron N_e amounts to:

$$i_e = \sum_{s=1}^{N} W_{es}(t) \cdot x_s(t) . \quad (5.3d)$$

If none of the incoming pulses overlap with each other, all synapses go to zero very rapidly so that the current i_e is very small, too. In case any two pulses overlap, the pulse to the left delivers a small current to the synapse to the right such that this synapse grows at first and then decays depending on the value of r.

When r is so large that the synapse is growing exponentially all the time, the current i_e will be quite large. With r sufficiently small, the phase of exponential rise is negligible compared to the phase of exponential decay, and the current will amount to about $2 \cdot w \cdot x$. Hence, r determines how many pulses

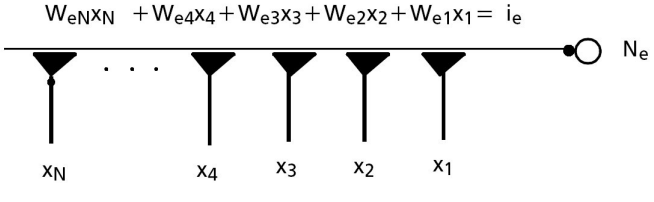


Fig. 5.19 Synchronisation detector ($\mu < 0$)

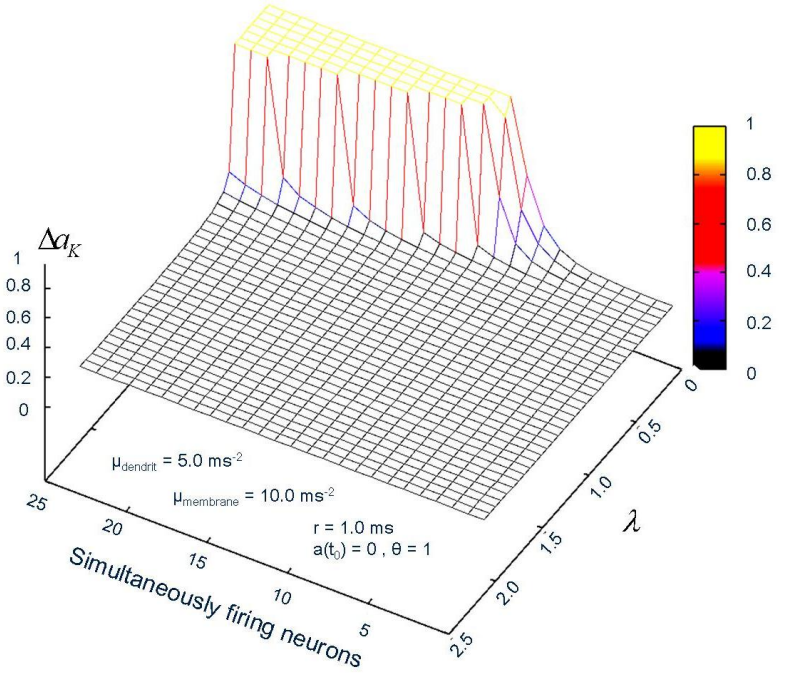


Fig. 5.20 Membrane potential as function of λ and number of simultaneously firing neurons

have to appear simultaneously before the exponential rise of the current can occur. By integrating this current with a dynamic synapse, which is situated at the membrane, has an adjustable threshold $\lambda \cdot \theta$ and obeys the equation

$$\frac{dW_e}{dt} = -\gamma(W_e - w)(t) + \mu \cdot \left(\int W_e(t') \cdot i_e(t') dt' - \lambda \cdot \theta \right) \ , \ \mu, \lambda > 0 \ , \quad (5.3e)$$

neuron N_e will create a pulse provided a fixed number of neurons as determined by r und λ have fired simultaneously.

Figure 5.20 shows the membrane potential as a function of λ and the number of simultaneously firing neurons. For $r = 1\,ms$, $\lambda = 0.05\,ms$ every pulse of the synchronisation detector indicates that at least 10 recognition neurons have fired synchronously and overlapped for at least $0.5\,ms$.

5.6 Time Coding and Correlation

Because synchronisation means an almost simultaneous firing of pulses thereby indicating signals of equal strength on the input side, synchronisation can be viewed as a particular case of time coding. Time coding, as well as pulse

coding, uses sequences of pulses, ie. pulse rates to indicate the respective event. Time coding is different from pulse coding only by the fact that all neurons synchronise their pulse activity. This pattern formation is the new quality which pulse coding does not possess.

Whenever an averaging of the input signals is required, pulse coding with constant synapses is appropriate. When a temporal-spatial correlation is to be detected, time coding with dynamic synapses is the right choice. As the experiments in this chapter have shown, time coding does not supersede pulse coding, rather both coexist: pulse coding as well as time coding each represent a dimension of coding the information. This is expressed clearly in Figure 5.2c.

5.7 Entropy and State Space

Distributions and entropies result for constant as well as dynamic synapses. With dynamic synapses, the entropy is no longer the only information carrier, but there is also the correlation of firing times of pairs or larger sets of neurons. Fundamentally this correlation is controlled by the input signals, in our case, pictures. There results a state space Z, the number of partitions of which is ruled by the image under consideration. Please note that the state space still gives rise to an observable statistics, thus entropy and distribution both exist.

Therefore, the definition of the transfer characteristics of a net with dynamic synapses requires details about the volume of the state space and the entropy. Accordingly the term „information-carrying function“ must be extended to the mapping of the pair (Z_{in}, E_{in}) on the input side into a pair (Z_{out}, E_{out}) on the output side[3]. Of course, this extension is applicable to nets with constant synapses. Since the state space is the same maximal set of all partitions on the input side as well as on the output side we did not mention it before.

For example, let us consider a fully connected net with $10 \cdot 10$ neurons. Then all neurons have started to fire synchronously within $0.013\,ms$, provided all neurons view the same input signal (Figure 5.21). Likewise for the finishing of the transmitting process. Instead of 2^{100} partitions, only a few hundred have to be dealt with. This shows a considerable reduction in the volume of the state space. Furthermore, the experiment teaches us that the temporal resolution of the events in question can easily undershoot the duration of a pulse by 2 orders of magnitude.

Let us next consider a net of 11 fully connected neurons and study the dependency of the synchronisation on the set of initial values. As we know from experiments with constant synapses (Figure 2.12) the initial values influence

[3] As a picture can be decomposed into a set of subsets featuring constant pixels (detected by spot detectors) and constant gradients of pixels, respectively, (detected by special feature detectors and subsequent spot detectors as shown in chapters 6-7) the above extension holds for each subset. When speaking of the information contained in a picture we must always refer to this decomposition.

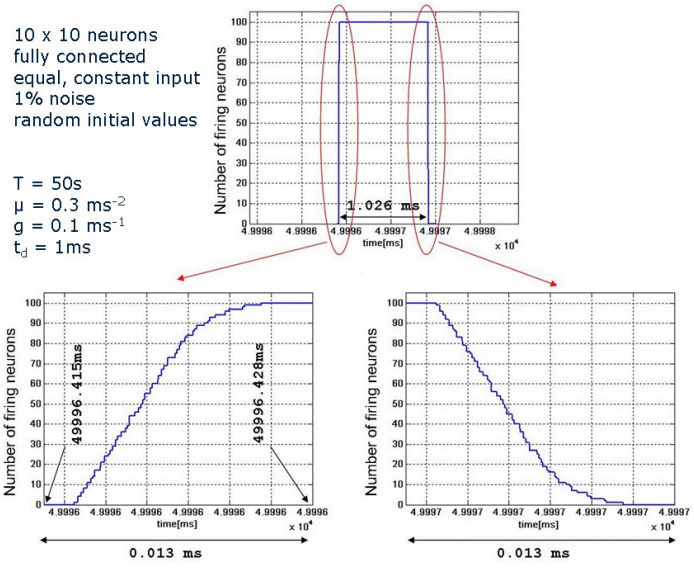


Fig. 5.21 Synchronisation process is about 80-times shorter than duration of a pulse

the distribution of partitions quite considerably. Figure 5.22 shows that in the case of dynamic synapses, the influence of initial values is negligible, as may be expected.

The net is started from 3 different sets of initial values (left without noise, right with 10 % noise), and the normalized activity histogram is recorded. Ideally, without any noise and equal input for all 11 neurons, there would be two partitions only: the one exclusively receiving and the other exclusively trans-

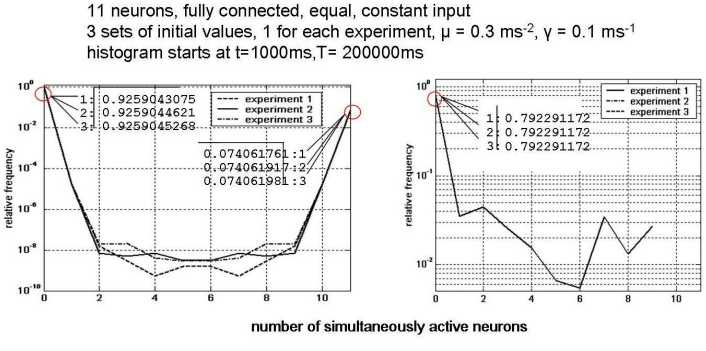


Fig. 5.22 Frequency of simultaneous occurrence of 1 to 11 transmitting neurons without (left) and with 10 % noise(right)

mitting. This situation is approximated for only 11 neurons on the left part of Figure 5.22 quite well: all pulse activity except for 0.01 % coincides with the ideal one. On the right part of Figure 5.22 no partitions with 10 or 11 transmitting neurons exist when 10 % noise is present in the input. After all, the 500 partitions with 7 to 9 transmitting neurons still represent about 50 % of the total transmitting activity (to be compared with almost 100 % for 11 neurons and without noise). As a consequence, synchronisation is clearly visible.

5.8 Preliminary Considerations on the Statistics of Synchronisation

Nets with excitatory dynamic synapses respond to correlations on the input side by synchronisation of the pulse activity: the state space of all possible partitions shrinks enormously, and the entropy falls to a lower value. Conversely, if the input image does not contain correlating pixels, ie. does contain noise only, the state space is of maximal volume and the entropy is high. Since the transition from one picture to the other implies always a sudden change in correlation content, the stationary theory determined by entropy maximization, as developed in chapter 3, is no longer sufficient. Instead, a dynamic theory of the distribution of partitions is needed. Since there is no reason to change the signal energy - the one used so far is the source of all the dynamics and, in particular, of the synchronisation phenomena - the properties of the solution space of the original signal equations, and thus the state space, must be investigated. From this we can obtain the equation of the distribution determining the temporal change of the volume of the state space.

The theoretical analysis can be formulated integrally in the asymptotic case of fully settled synchronisation. Then, the pulse waves are observable and tell directly which partitions are forming the state space; this information has to be taken into account as a side condition for the normalisation (cf. the case of the entropy S_1, chapter 3, (3.12)-(3.14)). Anyway, the calculation grows quite complicated and tedious so that we spare the reader the exemplification of the theory unless we are invited to do so by the application of nets with dynamic synapses dealt with in chapters 6-9.

5.9 Summary

Since spatial pattern recognition by nets with constant synapses can not work because on the membranes of the neurons the mixing of the signals destroys the information, a net dynamic must be constructed which reproduces the interrelationships present locally at the side of the input.

The dynamic synapses we introduced rise or fall rapidly and are able to generate temporal-spatial correlations of the pulse activity of neurons, provided those exist on the input side. This phenomenon is unique in the sense

of cause and effect. This way correlation appears as a further dimension of information, besides the one of entropy.

In the theoretical description the correlation can be accounted for by the shrinkage of all partitions in the state space to a much smaller volume which still carries the inevitable stochastics. In the generic case of dynamic synapses, a partial differential equation for the distribution must be solved, depending on the correlations present on the input side.

The ability of the dynamic synapses to detect patterns appears in the segmentation of images into spots of equal brightness: connected sets of pixels of equal brightness or colour (or any other feature) will be detected as a unit by virtue of the synchronised pulse activity of the neurons. Here the pulse rate is proportionate to the value of the brightness in the spot. Since synchronisation is inherently independent of the brightness, the segmentation is robust. Synchronisation, thus, is characterised by the binding of equal things to a single unit. How different things can be bound together to an object, say, will be dealt with in chapter 8. However, it does not occur through synchronisation, although it uses dynamic synapses.

The versatility of the dynamic synapses also shows up in their capability to detect gradients of brightness, etc. Unlike the case of segmentation with only positive μ synapses, a mixture of both types and those seated at the dendrite of a neuron are needed. Whether at the membrane side or at the dendrite side - the differential equation for all synapses has the same structure. The one is controlled by voltage, the other by current.

Nets with only negative μ synapses destroy any correlation present on the input side, they decorrelate. With the right mixture of positive and negative μ synapses, gradients can be detected and nets be constructed which put out the difference of the pulse rates present at two inputs.

A second important question - what are the basic types of calculation for neural nets - can therefore be given an initial answer. There exist:

1. the structureless mixing of information by nets with constant synapses,
2. the synchronisation of nets with excitatory synapses as an indication of a spatial correlation and pattern, respectively,
3. the generation of pulse differences and sums,
4. as well as the detection of coincidence and anti-coincidence, resp., by nets with correlating and decorrelating synapses, resp.

As will be shown in the next chapters, these types of calculation are sufficient to build all the stages of the object recognition chain, in particular feature and object recognition.

6

Nets for Feature Detection

Arne Heittmann and Ulrich Ramacher

Feature detectors can be constructed by using the basic computation rules of pulses presented in the last chapter. For instance, for a straight edge some number of receptors can be arranged to the right and left of the edge (Figure 6.1).

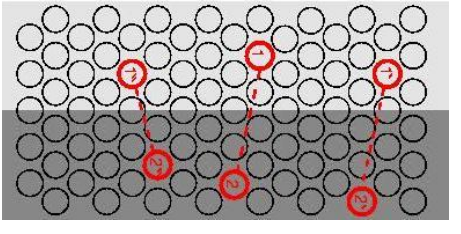


Fig. 6.1 Receptors seated left and right of the straight edge

Let us form pairs of receptor neurons by selecting two receptors situated at about the same height to the left and right of the edge. Each pair constitutes the input pair from which the pulse difference is computed as shown in Figure 5.8. In this context we call the pulse difference net a gradient detector.

How must the single gradient detectors be arranged such that the pulse rate of a single neuron indicates the presence of the edge? A minimal requirement is that the pulse rate must increase with the number of active gradient detectors. As more gradient detectors respond, the more the set of possible intensity distributions is narrowed down and the probability increased that a straight edge is present. Additional requirements may call for a higher weight of those gradients situated close to the edge and lower weight for those at the border. In this book we will be using a simple homogeneously-weighted connection of the single detectors (Figure 6.2a). Here, the symbol from Figure 5.11 is used as a compact description of Figure 6.2a.

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 105–118, 2010.
© Springer-Verlag Berlin Heidelberg 2010

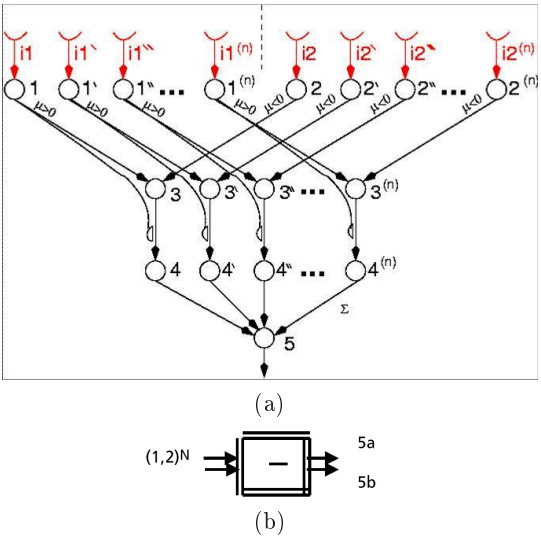


Fig. 6.2 (a) Simple detector for a straight edge (b) symbol for a symmetric gradient detector with N input pairs

Usually the gradient detectors can be attached to the receptors of the retina in an arbitrary order. In addition to edges with a preset direction, also curves, segments of circles or any specialised forms can be dealt with by the appropriate selection of the receptors, such that neuron N_5 responds particularly well.

This way, we could imagine many features which are to be detected, the sole question being how many of them would be necessary. As a matter of fact the world of real images is full of variety. There are textures, edges and curves of arbitrary shapes and locally varying gradients, to name a few of these examples. The variety of structures of images must, therefore, be reflected in the variety of feature detectors, in order to extract sufficient information from the visual scenes for the imaging system.

From the point of view of the *computer vision discipline*, the answer to the above question about the number of features would depend on the specification of the world(s) of images and would mean having to solve an optimisation problem: to find the minimal number of features derived from some subset of the world of images under consideration such that the differentiation of objects is maximized. Depending on the choice of the image subset, the robustness of the solution would turn out to be more or less satisfactory. Accordingly the determination of the local distribution of receptor pairs needs to be defined, which would look quite different from the one of Figure 6.1. As we are in the rare position of using experimental findings about the visual system of mammalians, we don't go down this path.

Instead, we will first give an overview of the basic anatomy of the first layers of the visual system of mammalians and give the most important hints about the information processing in these layers by means of Gabor wavelets. The rest of the chapter discusses the construction of a neural net for arbitrary Gabor wavelets and reports the measuring of their properties.

6.1 Overview of Visual System

The conversion of light into electrical signals and a set of pulses occurs in the retina. Figure 6.3 shows the five cell types (A) and their interconnection (B). At first incidental light is converted by photoreceptive cells into continuous electrical signals which causes a so-called hyperpolarisation of the membrane (Figure 6.3(B), cell upper left). Synaptic connections pass on the corresponding signal to horizontal and bipolar cells, the local connection of which results in an amplification of the local contrast of the image. The pre-processed image information is finally transferred to the *ganglion* cells which produce the pulses or spikes.

The axons of the ganglion cells form the optic nerve (nervus opticus). The optic nerve forwards the information which is encoded as a sequence of pulses to the lateral geniculate nucleus (LGN, Ganglion geniculatum laterale) which acts as a switching substation between the retina and the primary visual cortex (Figure 6.4).

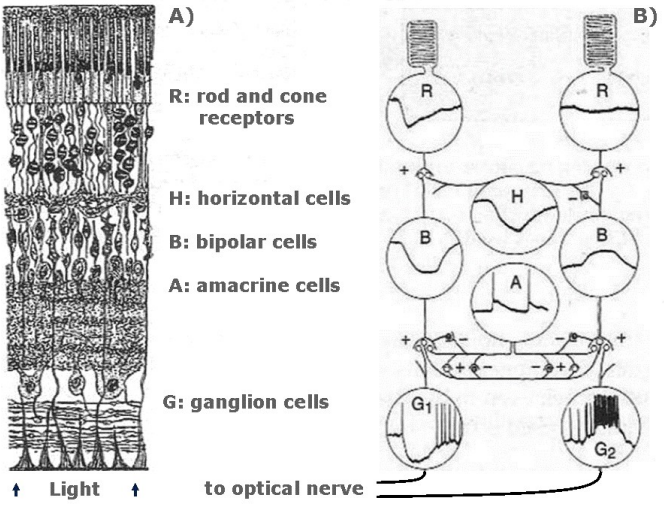


Fig. 6.3 Cut of the retina (A) and local interconnection of horizontal and bipolar cells (B) [44]

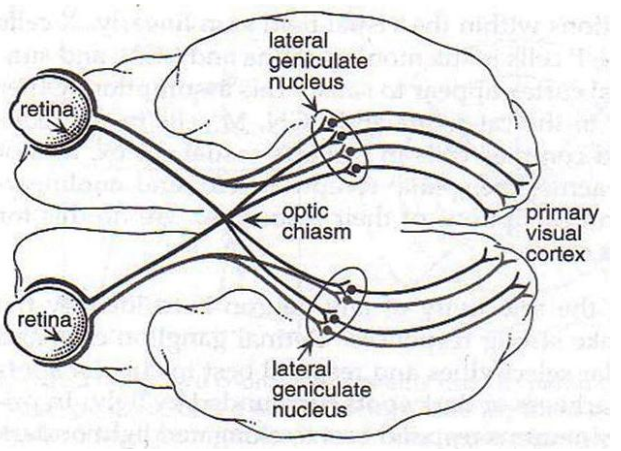


Fig. 6.4 Nervus opticus and ganglion geniculatum laterale [44]

Part of the two optic nerves crosses at the optic chiasma, whereby the information of both eyes is present in each brain hemisphere. The neurons of the retina, the LGN and primary visual cortex react to stimuli placed at local domains of the retina. These domains are called receptive fields. Those stimuli placed outside a receptive field of a given neuron don't generate any response from the neuron, whereas a stimulus placed inside the receptive field causes a significant reaction.

The neurons of the retina, the LGN and the primary visual cortex are divided into two classes. When stimuli placed at diverse locations of a receptive field cause its neuron to respond by linear overlaying of the responses of the single stimuli, a cell is called linear, otherwise nonlinear. Examples of the class of linear cells are the x-cells of the cat's retina, the p-cells of the retina and LGN of primates as well as *simple cells* of the human primary visual cortex. Examples of the class of nonlinear cells are the Y-cells of the cat's retina, the M-cells of the retina and LGN of primates as well as the *complex cells* of the human primary visual cortex.

6.2 Simple Cells

Let us focus next on the so-called „simple cells" (sC) of the primary visual cortex. The receptive fields of the sCs are comprised of small bounded fields of the retina (Figure 6.5).

sCs prove to be sensitive as to orientation and spatial frequency of the stimulus presented. In the early sixties these properties were considered to be incommensurable, and consequently sCs imagined to lead to spatial frequency filters or orientation detectors. Both functions, however, can be covered with

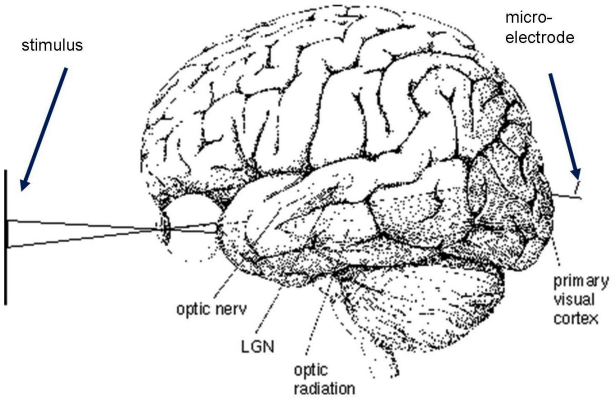


Fig. 6.5 Location of „simple cells“

a unique model. In the mid seventies Maffei and Fiorentini recognised that the response behaviour of sCs is linear [45]. Full clarification was obtained by Jones and Palmer in the year 1987 [2–4]. In their experiment light pulses - some bright, others less - were applied to the retina and the number of pulses produced by a single sC recorded as a function of the location of the stimuli (Figure 6.6).

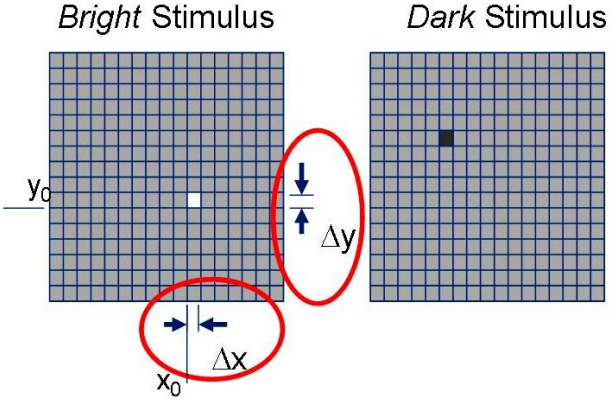
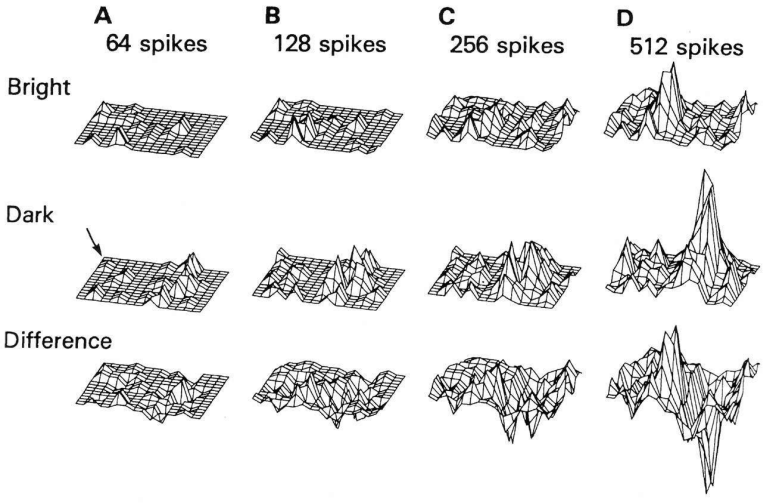
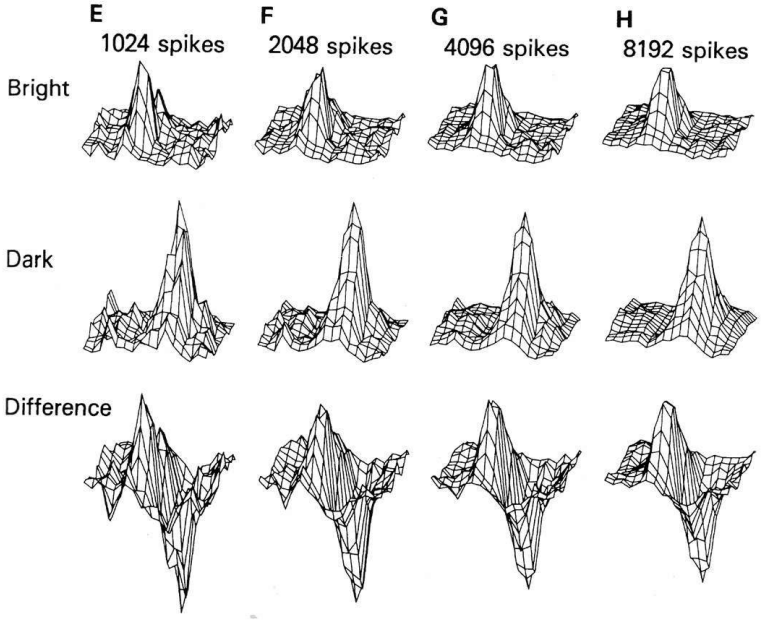


Fig. 6.6 Locations of stimuli on the retina for bright and darker light pulses

In Figure 6.7a,b the contours are displayed as a function of the number of stimuli applied over time. When the response obtained with the darker light pulses is subtracted from the response obtained with the lighter light pulses, the contour of a Gabor wavelet results.

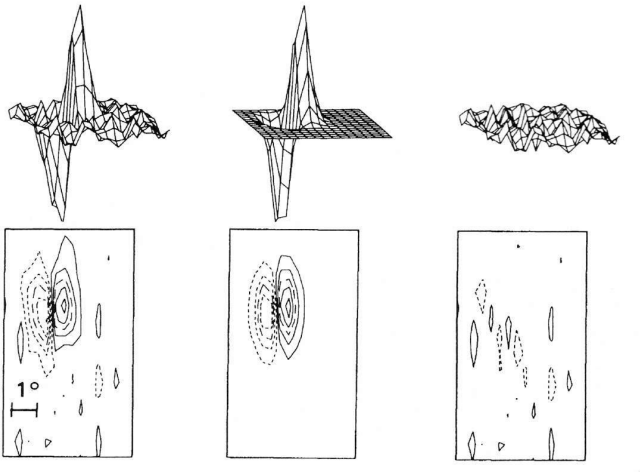


(a) 64-512 spikes



(b) 1024-8192 spikes

Fig. 6.7 2D histogram of the number of pulses produced by a sC, taken for different numbers of stimuli applied over time, and separately recorded for bright and darker stimuli [2].



(c)

Fig. 6.7 Example of a measured and a mathematical profile as well as the error profile [3].

36 sC of cats and their corresponding 2D spatial as well as frequency profiles were measured in detail. From that it became clear that measured profiles and the Gabor wavelet coincided and were statistically indistinguishable (Figure 6.7c).

Of course, these experiments did not reveal the neural architecture of the Gabor wavelet behaviour. For the rest of the chapter we dedicate ourselves to the task of reconstructing the measured input-output relation for Gabor wavelets by means of a neural net.

6.3 Creation of Detector Profiles for Gabor Wavelets

Linear filters can be realised with a classical analogue neural architecture. In doing so, the continuous transfer function is replaced by a sampled discrete version. The filter coefficients produced by the sampling are interpreted as continuous synaptic weights. Each of these synapses contacts a pixel from the receptive field, which produces an analogue signal $I(x, y)$. Multiplied by the synaptic weight, the input results in a current which is accumulated at the membrane of the neuron. The analogue signal response of the neuron follows a sigmoidal function and is interpreted as a mean pulse rate. When the linear part of the sigmoidal is used, a linear input-output behaviour results, ie. the neuron and the synapses result in a linear filter. Note that positive as well as negative synaptic weights have to be used so that the filter features a zero mean.

This perspective leads to the implementation of filter cores engineered by the classical filter theory and image processing over decades. In our opinion this implementation may secure the measured input-output characteristics, but it neither relates to the pulse signal representation of neural nets nor discloses how neural nets process information by means of pulses. In the following we present a net that very closely generates a Gabor wavelet response and uses only positive values for synapses.

We describe next the characteristics of a detector by means of the local intensity distribution to which it is particularly sensitive. Three basic questions have to be answered:

(i) are there any characteristics that describes the essential properties of the detector?
(ii) which operation transfers the signal $I(x, y)$ into pulses?
(iii) which particular brightness distribution allows the detector to respond maximally ?

In the theory of linear systems the characteristics of a filter are described by the filter core G. Since the pulse neurons considered in this book are nonlinear, it is not possible to apply this theory directly. Let us briefly re-examine the linear theory before dealing with the nonlinear case. G is a (continuous or discrete) function of the location variables x and y. The filter response g at the location (x_0,y_0) of the retina R is created by revolving the filter core G with the intensity distribution I:

$$g(x_0, y_0) = \int_R I(x, y) \cdot G(x - x_0, y - y_0)\, dxdy \ . \tag{6.1}$$

In case of a locally bounded receptive field of a detecting sC, it suffices to perform the convolution over the perceptive fields only. When the filter core and the intensity distribution are equal, then the filter response becomes maximal, as is well known.

For a given filter (or detector) the characteristics G is determined by sampling the receptive field. In the introductory section about the detector profile of the sCs, this method was used. The zero mean feature of G implies for a homogenous intensity distribution B:

$$g(x_0, y_0) = B \cdot \int_R G(x - x_0, y - y_0)\, dxdy = 0 \quad \text{for all } (x_0, y_0) \ .$$

If we choose in a further experiment

$$I(x, y) = B + (H - B) \cdot \delta(x - x_1, y - y_1) \ ,$$

ie. if the intensity distribution equals a single spot of brightness H at (x_1, y_1) and is surrounded by a background intensity B, then the filter response results in:

$$(H - B) \cdot G(x_1 - x_0, y_1 - y_0) \ , \tag{6.2a}$$

and for a darker spot of brightness D:

$$(B - D) \cdot G(x_1 - x_0, y_1 - y_0) \ . \tag{6.2b}$$

By means of this experiment the filter core G can be read off point by point and the characteristic intensity distribution be determined, which makes the filter response maximal.

Analogue thoughts hold for the pulsed detector response of Figure 6.2a-b. Let us assume that the detector possesses a characteristics G. Firstly, the zero mean feature of the detector must be guaranteed. This can be assured if the gradient detector shown in (Figure 5.8 and 6.2a) is used and followed by a symmetrical difference stage (cf. Figure 6.8, cf. also Figures 5.11, 5.15). The remaining offset is considered to be noise.

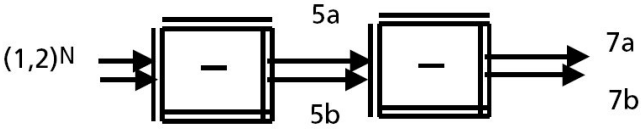


Fig. 6.8 Symbol of the feature detector

When a small region (rectangle in Figure 6.9, reference point is in the lower left corner) of the receptive field of a single sC is illuminated homogeneously with brightness H and the background with B, the pulse rates of the neurons $N_{5a,b}$ and $N_{7a,b}$, resp., can be analysed.

We assume that the illuminated rectangle contains no pairs, but only one input neuron of a pair. Thus, the light spot used is smaller than the mean

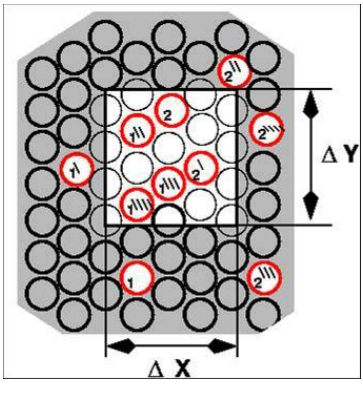


Fig. 6.9 Sampling of the receptive field of a *simplecell*

distance between the inputs of a pair and larger than a photoreceptor of the retina. To first order the pulse rate produced by the gradient detector (Figure 5.8) is given by $(H-B)+noise$ at N_{4a} (if input 1 lies in the rectangle and input 2 outside) and by $(H-B)+noise$ at N_{4b} (if input 2 lies in the rectangle and input 1 outside), respectively. In both cases, mean values are available at N_{5a} and N_{5b} in Figure 6.2a-b, respectively.

Let a feature with an arbitrary orientation be present in the rectangle. A unique result is produced by $N_{5a,b}$ according to Figure 6.8: the one output represents noise, the other output a pulse rate which is proportionate to the mean of the pulse differences present at the input side. Let us designate by $n_1(x,y)$ and $n_2(x,y)$ the corresponding local densities of the inputs 1 and 2 on the retina. Then, the pulse rates of neurons N_{5a} and accordingly N_{5b} read:

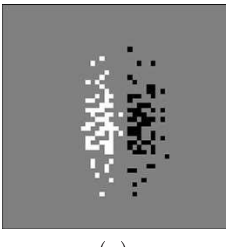
$$(H-B)\cdot\int dxdy\ \ n_1(x-x_0,y-y_0)\ , \tag{6.3a}$$

$$(H-B)\cdot\int dxdy\ \ n_2(x-x_0,y-y_0) \tag{6.3b}$$

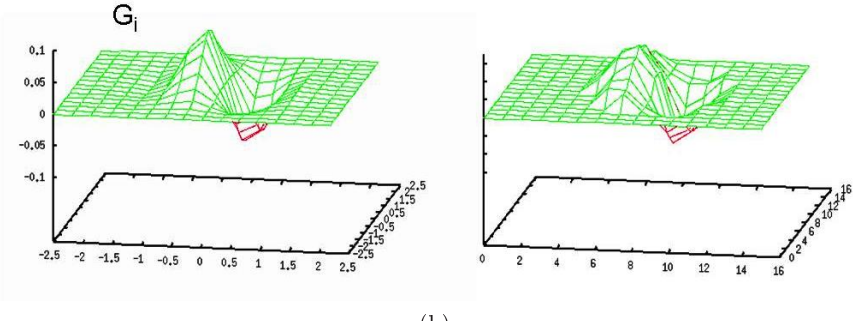
and, correspondingly, for N_{7a}, N_{7b} :

$$(H-B)\cdot\int dxdy\,(n_1-n_2)\,(x-x_0,y-y_0)\ , \tag{6.3c}$$

$$(H-B)\cdot\int dxdy\,(n_2-n_1)\,(x-x_0,y-y_0)\ . \tag{6.3d}$$



(a)



(b)

Fig. 6.10 (a) receptor densities for a dedicated Gabor wavelet; (b) Gabor-Wavelet (left) and its pulse rate representation (right) $(N_{7a}-N_{7b})$

Let us assume that N_{7a} generates the relevant pulse rate. Since the experimental determination of the pulse rate profile resulted in Gabor wavelets, we should choose:

$$G(x - x_0, y - y_0) = (n_1 - n_2)\,(x - x_0, y - y_0)\ . \tag{6.3e}$$

This way, an exceptional agreement is obtained (Figure 6.10).

In summary the densities n_1, n_2 determine the filter profile G (up to a normalization factor); in particular, it can be chosen to generate Gabor wavelets. Conversely, a targeted Gabor wavelet determines the densities.

6.4 Experimental Check

We select three different stimuli: edge, curve, and rotated edge (Figure 6.11). The stimuli cover 17×17 pixels and are moved/rotated over the detector. For the displacement $a = 0$ the zero mean intensity profile equals the above detector profile G. The pulse rates of neurons N_{5a} and N_{7a} of the Gabor wavelet detector according to Figure 6.8 were measured as a function of the intensity distribution on the input side. Figure 6.12a displays the pulse rates for neuron N_{5a} (left) and neuron N_{7a} (right) in the case of the edge, the blue curve is obtained with the ideal Gabor wavelet for various displacements. The case of the curve is shown in Figure 6.12b, and the results for the rotated edge ($0 < a < 180°$) are presented in Figure 6.12c.

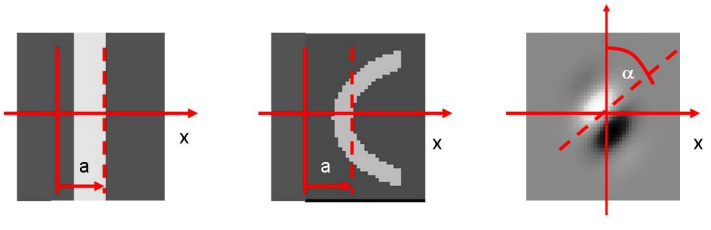
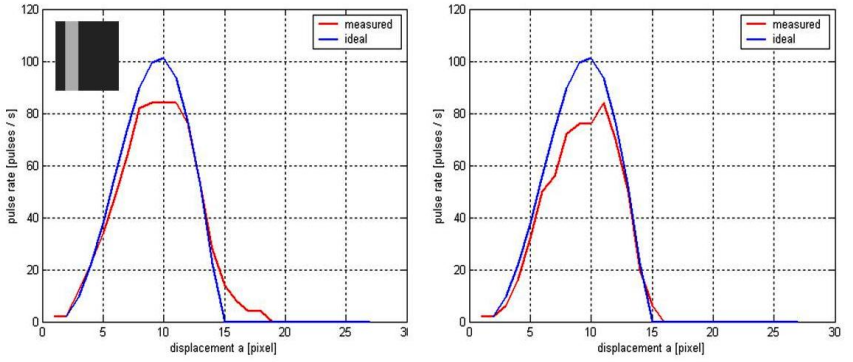


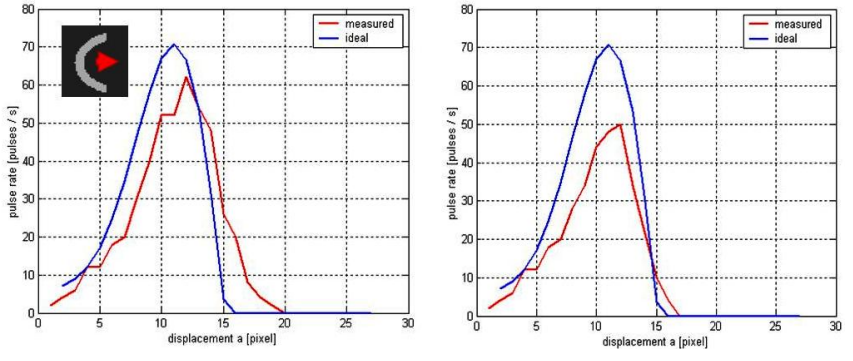
Fig. 6.11 The three selected stimuli

The gradient of the pulse rate of neuron N_{5a} is always smaller than the ideal curve, but the curves of N_{7a} exhibit almost the same gradient as the ideal curves. Once more this confirms the effectiveness of the downstream difference stage.

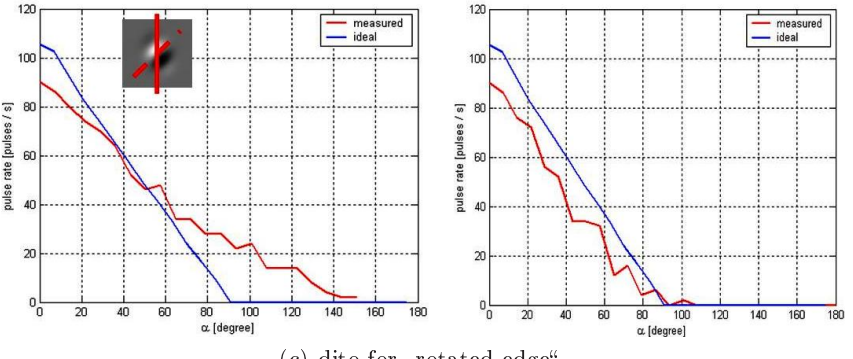
Representative of many other experiments with Gabor wavelets, Figure 6.13 shows the results for two complex Gabor wavelets, one being orientated horizontally and the other vertically. The first row displays the mathematical ideal Gabor wavelet response, the second row the neural detector's response (real and imaginary parts separately).



(a) Ideal (blue) and measured (red) response for „edge“ at N_{5a} (left) and N_{7a} (right)



(b) dito for „curve“



(c) dito for „rotated edge“

Fig. 6.12

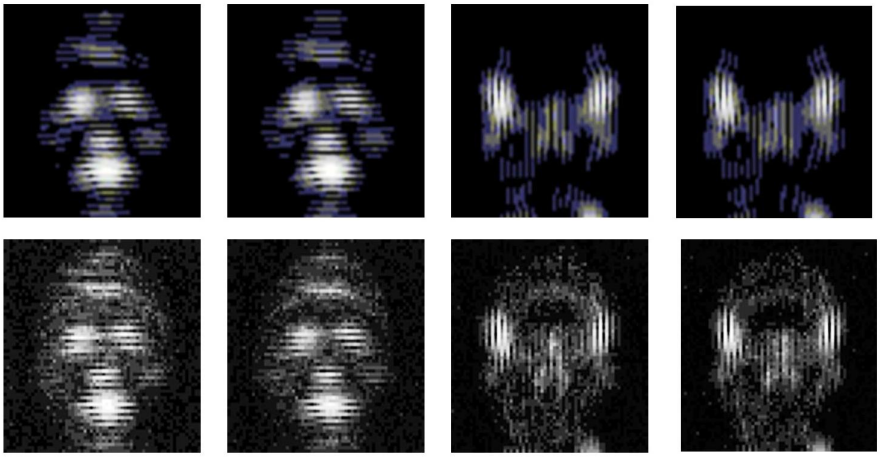


Fig. 6.13 Ideal (top) and measured (bottom) response of a horizontal and vertical, resp., complex Gabor wavelet

6.5 Summary

As *simple cells* of the primary visual cortex feature characteristics of a Gabor wavelet type, a neural net has been developed which is parameterisable and may produce any type of Gabor wavelets. The first parameterisation is comprised of the two densities n_1 and n_2 which tell how many photoreceptors are contacted in the receptive field by synapses of a *simple cell*. The synapses are grouped into pairs, and each pair presents their two pulse signals to the pulse difference net described in chapter 5. The second parameterisation accounts for the type of averaging of the responses of the neurons N_4 by the neuron N_5. For the purposes of this book a uniform synaptic weighting is sufficient, but any other distribution can be implemented.

The choice of the densities n_1 and n_2 mainly determine the characteristics of the feature detector. In this manner *neural* Gabor wavelets can be used very close to the mathematical ones. Please note that the Gabor wavelet detector uses dynamic synapses to compute the local gradients, thus involving a time-ordered relation, which is destroyed by its inherent averaging procedure, however. Hence, the internal time coding is not presented to the exterior world.

This book deals with imaging only; however, we expect that the gradient detector presented is quite universal and can be applied broadly, because the subtraction of data or the gradient of data must be a common task encountered in information processing in human brains.

Clearly, the number of feature detectors depends on the task under consideration. How many are needed, how they have to be distributed over the

retina and how they are interconnected to detect an object is discussed in the next two chapters.

7

Nets for Feature Recognition

Ulrich Ramacher and Arne Heittmann

Let us consider an image with 1024×768 pixels and let it be covered by a grid of 69×85 Gabor wavelet detectors, such that the pulse rate of N_{7a} and N_{7b}, resp., is obtained at each node of the grid. Let each detector possess 128 sensor pairs. Moreover, 6 complex, ie. 12 real Gabor wavelet detectors are applied. At each grid node, the vector of the pulse response of the 12 output neurons of the respective detectors is recorded. A further grid with 18×22 nodes and 12 detector levels just covers a face, which is chosen for reference (Figure 7.1 mid), and is moved over the fine-grain larger net.

In doing so, a node in the second grid is realised as a neuron, with 12 constant synapses, each one connecting one of the 12 nodes from the first fine-grain grid (Figure 7.2). Each synapse takes a value which is conversely proportionate to the pulse rate of the neuron, provided the referenced face is presented as the image (Figure 7.1 left). All neurons in the second grid are connected to their neighbours by dynamic synapses with positive μ. Thus, the storage of the referenced face is reflected in the values of the input synapses, and the detection of the face happens by synchronisation of the dynamic synapses.

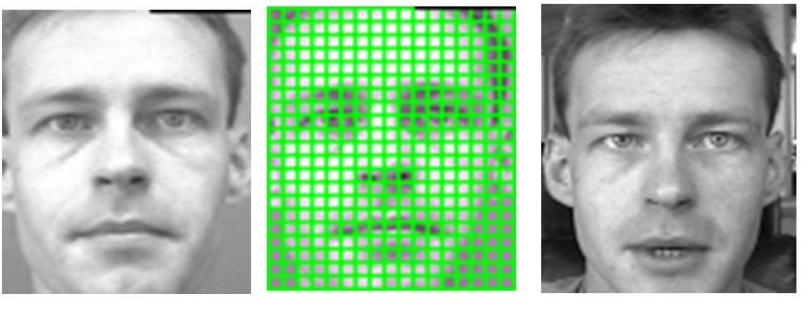


Fig. 7.1 Reference face, feature grids, and reference face with different look

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 119–139, 2010.
© Springer-Verlag Berlin Heidelberg 2010

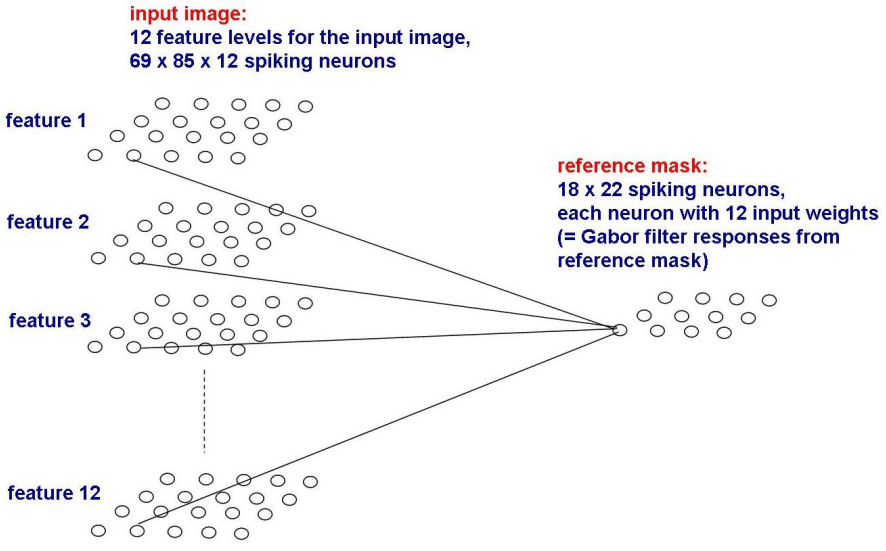


Fig. 7.2 Detector assembly for 12 Gabor wavelets

According to its construction, the second smaller grid is used to detect the correlation between the face presented as input and the reference face. For when considering the reference face, (maybe with a slightly different expression, cf. Figure 7.1 right) a synchronous pulse wave will appear at the place of best coincidence and, thus, the experimentally measured entropy drops suddenly (Figure 7.3).

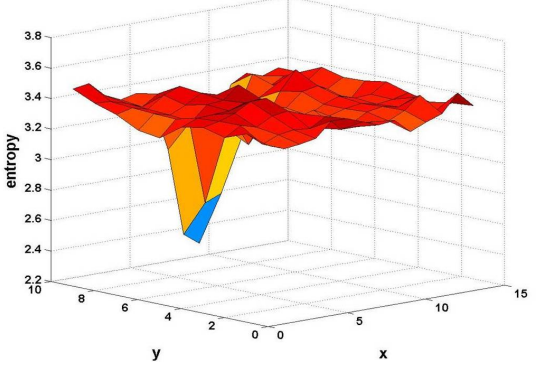


Fig. 7.3 Experimental entropy drops at the position of largest congruency

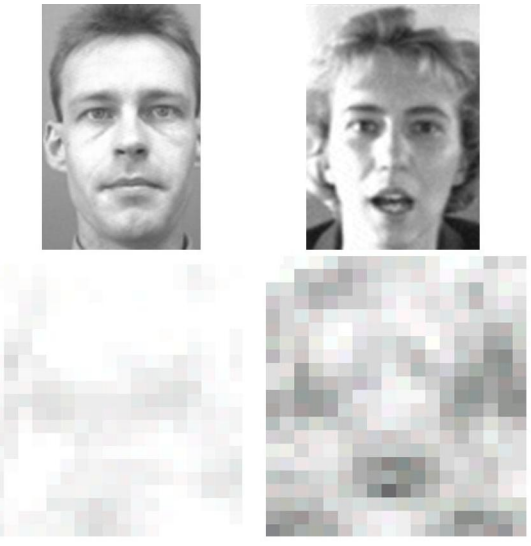


Fig. 7.4 Regions of differing pulse rates at the position of highest congruency indicate lack of robustness when a new face is considered (left reference face, right new face)

However, the robustness is inadequate. Looking at a different head, the position of best coincidence does not result in a pulse wave, but in many small regions of synchronisation (Figure 7.4). Hence, a head recogniser - not to mention a face recogniser - cannot be obtained by direct coupling of the feature detectors to the object. Note also that the detection would not be independent of the level of illumination.

7.1 Principles of Object Recognition

Let us consider any world of images - for instance, the world of head images - in which a fixed number of objects have to be recognised, and let us deconstruct each object (for example a head) into a number of non-overlapping object components (the brow or the nose, for instance). Let each object component define a zone where its component may appear in any form, and let so many different feature types lie in this zone as are needed for recognition of the object component. As each object may appear at different distances (relative to the observer), an object component could no longer be recognisable. Instead, the object (here the head) could convert to a component of another object (here the human body), or a change of perspective may cause a new deconstruction of the object into its components (turning from the forehead to the back of the head, for instance). Accordingly the recognition of objects is a construction task which should obey the following principles:

1. For a given world of images, each object should feature only so many object components and zones, respectively, such that recognition is assured.
2. Within each zone, there should only be as many different feature types as is necessary for the recognition of the object component to work. Similarly, the coverage of an object component by feature detectors of a single type should be only as large as is necessary.
3. Invariance as to distance, rotation or change of illumination should prevail in the image region in which the object components are still visible.
4. When some of the object components vanish and new ones appear, the architecture of recognition should be able to be extended by object components and new or more features.

These principles aim at a preferably compact, robust description of an object and anticipate the definition of the object by binding together its object components.

Human beings recognise moving objects in an invariant fashion and may attribute object status to something under consideration, provided that thing is of sufficient importance. Let us further suppose that the visual system is comprised of a pyramid of complete detection and recognition features. Then the selection of a minimal number of object components, feature types and features does not seem to be as compelling as the above principles demand. Just the contrary: during the assembly of the world of images, in particular during the first encounter with an object, the full extent of the object is not immediately and fully apparent. Hence the better strategy could consist of using the huge number of synapses and neurons to detect as many object components, etc. as possible. In consequence, the number of object components and features could be higher than necessary for recognising an object within a given world of images. Since the process of object definition is rarely an instantaneous one, but a temporally extended one, a mechanism of differentiation is required to take advantage of the continuous interaction with the „object-to-be“, and to single out the important features. At the finish of this differentiation process, a set of object components will be established that should be close to the ones postulated by the above principles. These static principles for the final fully-developed notion of an object have, thus, to be complemented by dynamic principles:

5. At the first encounter the object is captured with maximal resolution: each locally dominant feature of any type is understood to be an object component.
6. With further encounters and more interactions, object components may resolve or disappear and even unite to form a new object component.
7. Each object component is assigned a zone in which it appears alone and with any deformation it may adopt, provided the object is presented in the referenced position. Perspective rotations may happen within the zone as long as the object components remain visible.

8. Each object is assigned a maximal perspective region in which it may appear as a reduced image of the object at its referenced position.
9. Perspective rotations in reference position that cause object components to disappear or new ones to pop up complement the object description.

In the following we will be using the principles 1-4 and 7 as the exit point for the construction. Principles 5, 6 and 9 still await realisation as neural nets. In chapter 9 an architectural scheme is proposed which implements principle 8, but it is too large to be proven by simulation.

7.2 Net Architecture for Robust Feature Recognition

Since we would like to recognise a great number of objects, many Gabor wavelets will be needed throughout the retina even if objects are in a reference position and standard viewing distance only. Hence we imagine the detectors of each type to cover the retina constantly and completely (for more details cf. chapter 9). Since the image of an object (or an object class) will always vary in its exact position on the retina, even when viewed at a fixed distance, and because the distinctive object components overlap a little or not at all, each object component can be assigned as large a zone of occurrence on the retina as possible.

Let us consider, for example, the brow and eye section of a face: at this location there is no overlap with other prominent features of the face, even if the eyebrow or eye perform all their possible changes (Figure 7.5).

In this zone there is a multitude of detectors of different types, but we restrict ourselves to a single dominant type in the zone, ie. we model the object component with a single Gabor wavelet. Exactly 480 horizontal detectors of it lie in the 20×24 pixel large zone, each one possessing a receptive field of 13×13 pixels. A brow or an eye will therefore be covered by a single feature

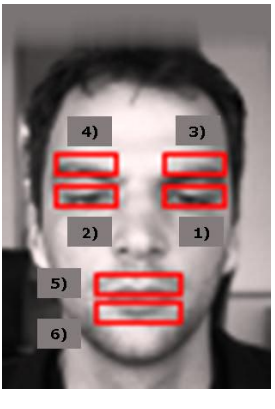


Fig. 7.5 Zones as maximal regions of appearance of an object component

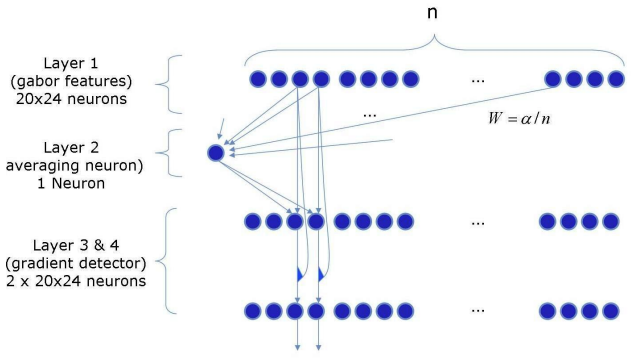


Fig. 7.6 Subtraction of average feature brightness from the original one

detector and zone, resp. The task is to determine the largest connected set of synchronously pulsing detectors in the zone. This requires several steps. First of all, the mean brightness of the zone is determined, and subtracted individually from the pulse rates produced by the detectors. For that purpose we use the pulse difference stage and note its asymmetry of computation: only when the pulse rate of the detectors is higher than the average will the neurons in layer 4 fire (Figure 7.6).

The layer 4 is succeeded by layer 5, which again consists of 20×24 neurons. These neurons are each connected by a synapse of the value 1/10 to the neurons of layer 4, directly situated above. Without any further connection, the neurons of layer 5 (also of layers 1 and 4) generate the activity displayed in Figure 7.7, recorded in each of the six zones as the number of neurons concurrently firing in a sliding 2ms window. Although Figure 7.7 does not indicate which ones of the 20×24 neurons fire, it is nevertheless clear that rarely more than 4 neurons fire, with the average being 2. This way we certainly cannot claim that a feature is present in its zone or not.

Let us recall, therefore, the functioning of the spot detector (cf. chapter 5, section 5.3) and regard the layer 4 as that layer which is connected by constant and equal synapses to the 20×24 neurons of a spot detector. Since the feature must lie somewhere in the zone, a respective set of neurons of the feature detector will fire with the same pulse rate and, hence, on average the attached neurons of the spot detector will all receive the same amount of pulse current. In consequence, depending on the connection scheme, a pulse wave will run over the spot detector neurons, or all neurons will fire synchronously. Anyway, the phenomenon of synchronisation will show up irrespective of where the feature appears in the zone (independence from location).

If an adjoining connection is chosen, only that feature will be recognised which perfectly matches the pre-set Gabor wavelet. With full connection, distant detectors of equal pulse rate but from different segmentation spots would appear as neighbours and, hence, spatial coherence be no longer a determining

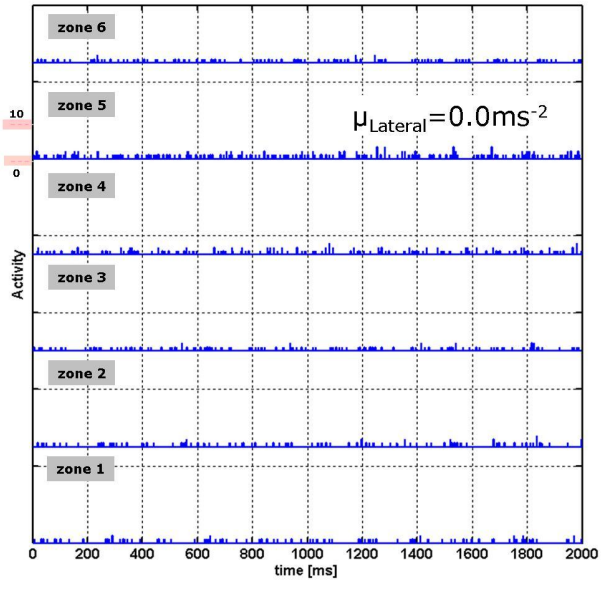


Fig. 7.7 Number of concurrently firing neurons of layer 5 (without lateral connection) as function of time

factor. Thus, the further optimisation must consist of the search for a suitable lateral connection of the neurons of the spot detector, such that all possible *native* appearances - and only those - of the brow and its Gabor wavelet, resp., are detected and recognised. This results in independence from deformations.

For the purposes of this book, a connection scheme is chosen for layer 5 such that each neuron is connected to its 4 horizontal neighbours on the left and right side, respectively (Figure 7.8).

Vertically, the neurons above and below are connected, as well as the neighbouring ones. Thus each neuron has 14 synapses, and in total 6100 dynamic and 480 constant synapses are needed for the field of 20×24 neurons in the layer 5.

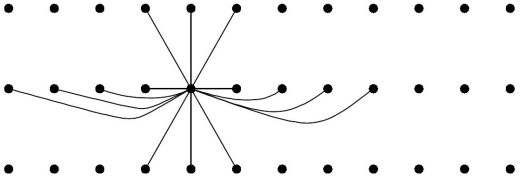


Fig. 7.8 Connection of each neuron in layer 5 (extract from the 20×24 field)

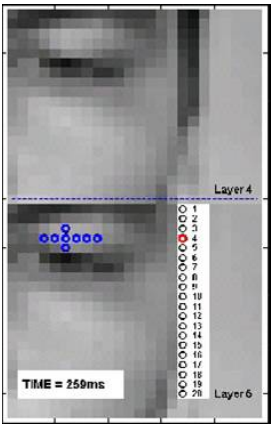


Fig. 7.9 Synchronisation in layer 5, with brow presented in reference position

Figure 7.9 shows an extract of the pulse activity for the eyebrow and eye zone, provided it is presented in reference position. It is noted that 8 neurons of the layer 5 are firing synchronously in row 4 (blue circles). The synchronisation is particularly impressive when seen in the video.

As discussed in the two previous chapters, the initiation of synchronisation depends on the number of inputs of almost equal pulse rate and the coupling factor μ of those synapses connecting the 24 neurons in a row. The larger μ is chosen, the fewer equal inputs are necessary, but the less resolution is obtained. Of course, the synchronisation is also a function of the connection scheme for the 480 neurons. With our choice of connection, we get useful recognition behaviour in the layer 5, when brightness or the form of the feature changes in the zone. Also, the orthogonality of different features is secured. Before we prove these properties, the layer 5 is complemented by a small net that indicates the state and extent of the synchronisation.

7.3 Feature Recogniser

Two synchronisation detectors (chapter 5, section 5.5) are added to each row of the field of 20×24 neurons in the layer 5, and connected to the 24 neurons of a row by means of dynamic synapses (cf. Figure 7.10). The synapses of the two dendrites differ only by the value of the coefficient r (cf. Eq. (5.3c)) and share the same value $1/2$ as the coefficient λ. The values of r are allocated such that just a single pulse causes the neuron N_2 of the synchronisation detector to fire, or at least 4 concurrent pulses cause the neuron N_1 to fire. The choice of the value 4 depends, clearly, on the size of a feature and the field.

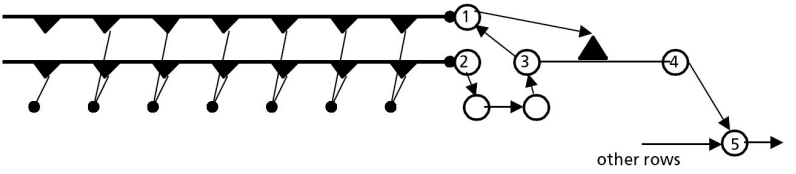


Fig. 7.10 Version of the synchronisation detector for unique activation (extract of the 20×24 field of layer 5, showing one row)

Two synchronisation detectors are necessary to prevent faulty activation, which would occur with a single detector. For, if the threshold of neuron N_1 is not reached because of fewer than 4 incoming pulses, N_1 may be pre-loaded such that, say, two subsequent pulses would cause a firing of a pulse of N_1. A corrective is provided by the arrangement shown in Figure 7.10. With the insertion of 3 neurons, the synapses of which are constant and of value 1.2, a pulse generated by neuron N_2 is delayed by $4\,ms$. The neuron N_3 transmits its pulse by a dynamic synapse with positive μ to the synchronisation detector's neuron N_1 and to neuron N_4 by a constant synapse of value 0.025. The synapse W_{13} obeys the common synapse equation,

$$\frac{dW_{13}}{dt} = -\gamma \cdot W_{13}(t) + \mu \cdot (a_1(t) - 0.5\ \theta)\ , \qquad \mu = 25\ ,$$

whereas the synapse W_{41} requires the dendritic version ($\mu = 50$):

$$\frac{dW_{41}}{dt} = -\gamma \cdot (W_{41} - w_{41})(t) + \mu \cdot \left(\int W_{41}(t') \cdot x_1(t')\, dt' - r \cdot 0.025\, x_3(t') \right)\ .$$

When neuron N_1 fires, it may mean that 4 concurrent pulses are present. Then neuron N_2 would pulse concurrently, but not N_3 because of the delay. Hence, the dendritic synapse W_{41} amplifies the N_1 pulse, and N_4 fires. The alternative meaning, as described above, would result in a pulse of neuron N_3 that causes the pre-loaded neuron N_1 to fire almost simultaneously, with the effect that the dendritic synapse is disabled and only the very low current $0.025 \cdot x_3$ is accumulated by N_4 for $1\,ms$. Thus, we have proven that neuron N_4 fires only when the feature is present in the row with 4 pulses. Neuron N_5 sums up the pulses of all rows. The whole arrangement reaching from layer 5 to neuron N_5 is called *feature recogniser*.

Let us now revisit our ideal brow and record the signal responses of the enhanced synchronisation detectors of the 20 rows (Figure 7.11).

For each row the number of simultaneously firing neurons in layer 5 is cited. Also, the pulse responses of neurons N_1 and N_4 are recorded. It is clearly visible that only 4 or more simultaneously firing neurons cause the synchronisation detector to respond.

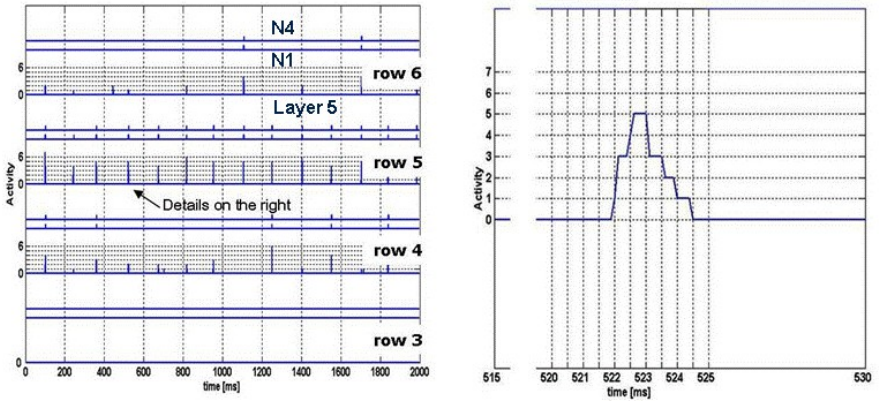
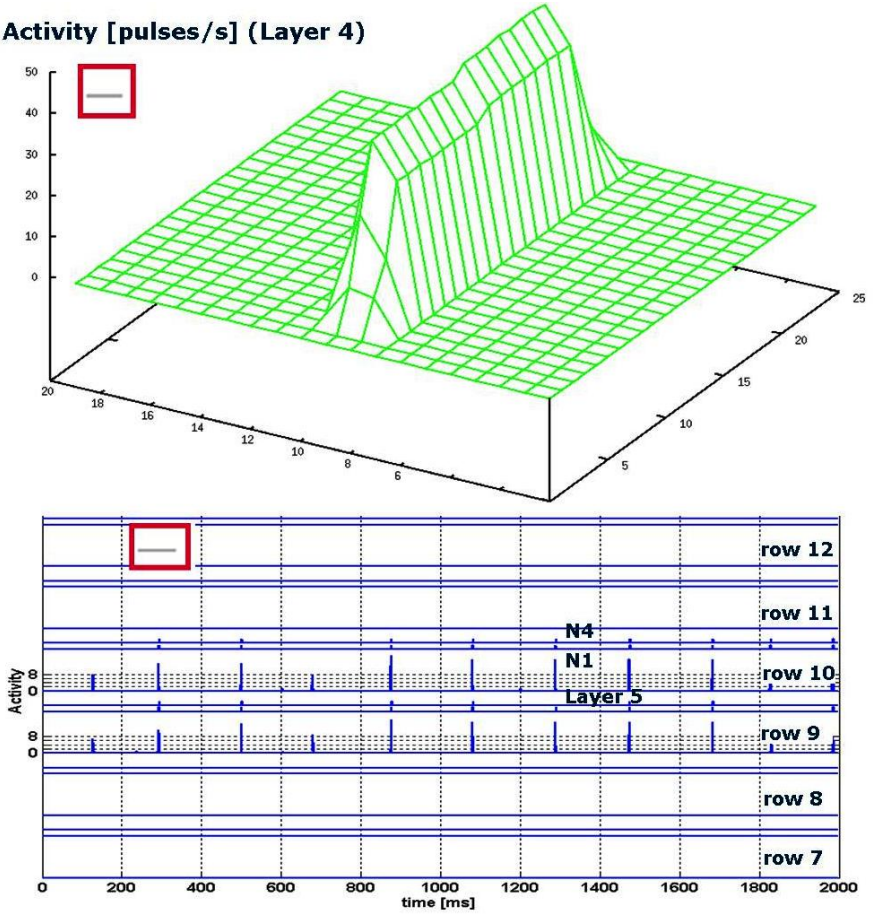


Fig. 7.11 Signal response of the the enhanced synchronisation detectors in case of the brow (active rows shown only)

7.4 Selectivity

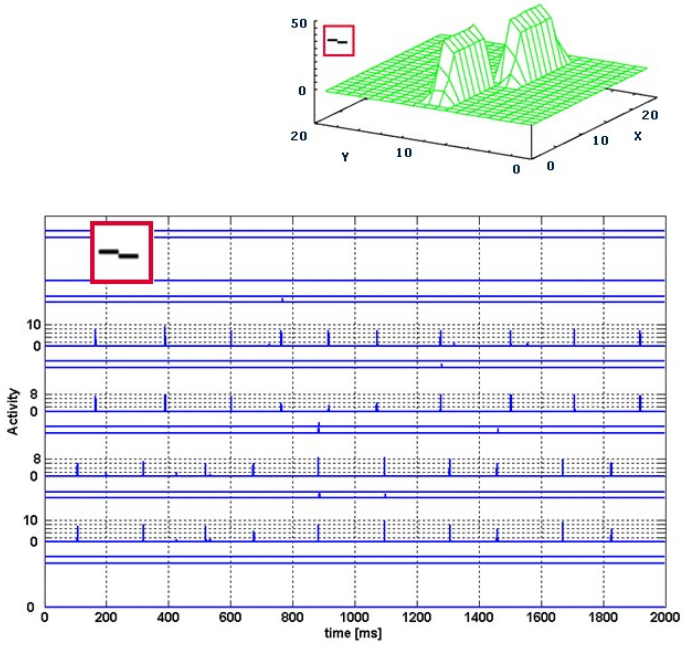
The choice of the local connection scheme in layer 5 determines how selective the feature recogniser will be. Let us consider a two pixel wide line, record the activity of the layers 4 and 5 (the latter including the synchronisation detectors) and then divide the line in halves which are slowly moving apart. The connection would turn out to be useable if it maps separated lines into separated activities such that the activity would be proportionate to the length of the line. Since the line was chosen to be longer than a brow, the threshold for one of the synchronisation detectors is set at 10 pulses, ie. 10 neurons must fire synchronously. The upper figure in Figure 7.12a displays the activity of layer 4, thus the pulse rate of the feature detector layer (relative to its average pulse activity). The lower figure shows that many neurons per row will fire synchronously in layer 5 (only those rows are shown which respond), with two extra lines on top showing the responses of the neurons N_1 and N_4. Obviously about 14 neurons fire simultaneously when the line is intact.

For the barely separated line there are about 8 neurons firing (Figure 7.12b), and this number stays the same when the two lines are moving apart (Figure 7.12c). The lines would no longer be separated, only if the vertical connection increases to a similar size as the horizontal one.

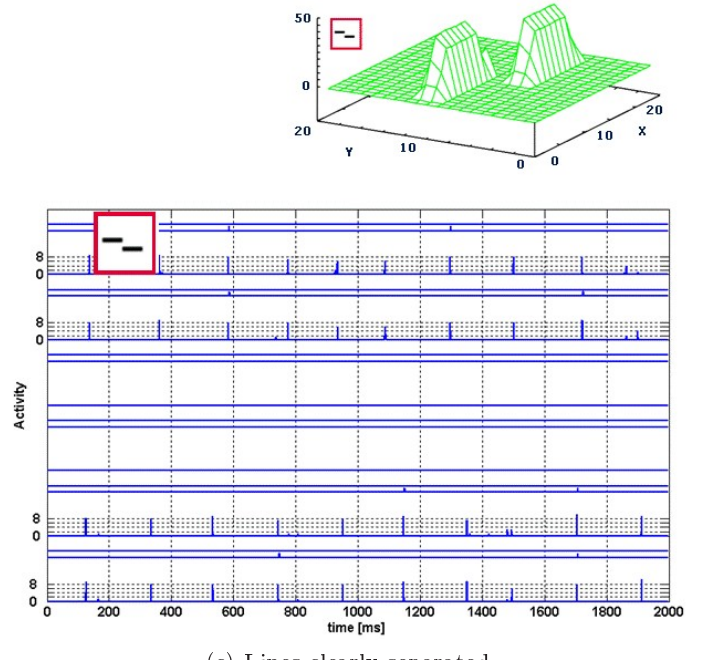


(a)

Fig. 7.12 Signal responses of the enhanced synchronisation detector for a line



(b) Line divided and barely separated

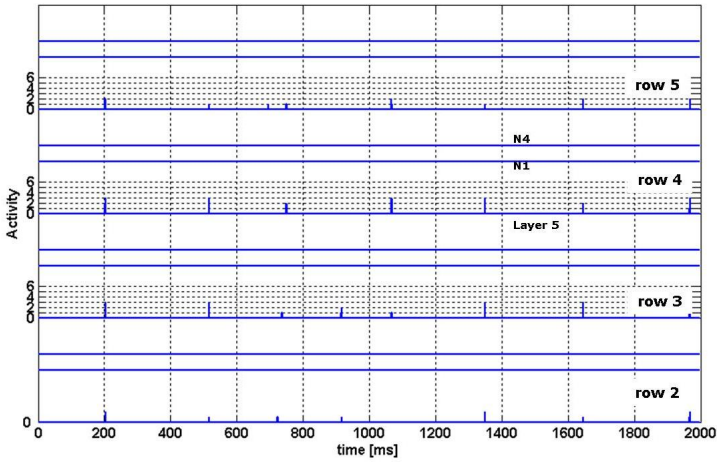


(c) Lines clearly separated

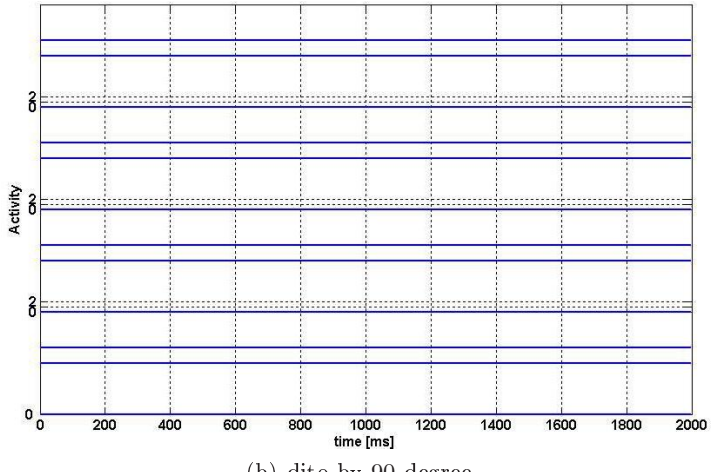
Fig. 7.12 (continued)

7.5 Orthogonality of Rotation

The selected feature under consideration should not generate a signal response after rotating the image by 90 degree. Let us rotate the head around the midpoint of the brow's zone. Then we expect zero response for a rotation equal or larger than 45 degrees. Figure 7.13a,b presents the behaviour: measured between 45 and 90 degrees neuron N_4 no longer responds. Figure 7.13a demonstrates beautifully that subcritical synchronisation in layer 4 does not cause a response from the feature recogniser. Hence, the connection scheme of layer 5 is convenient.



(a) Rotation of the brow by 45 degree

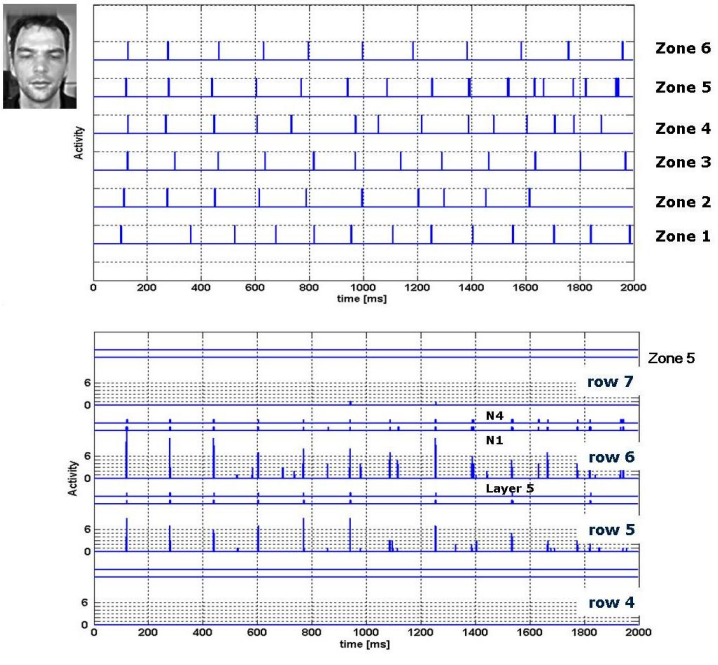


(b) dito by 90 degree

Fig. 7.13 Signal response of the synchronisation detector when rotated

7.6 Invariance of Function as to Brightness

Because it became apparent already in chapter 5, section 5.3, Figure 5.7a,b, it is not surprising that the synchronisation detector functions over a wide range of brightness. Let us consider a bright, a mid-bright and a dark image. For each zone, the signals from the rows are passed through synapses of value 1/20 to finally accumulate at the neuron N_5, which represents the zone's response for the feature under consideration. In this way, the three images produce responses as shown in Figure 7.14a-c. For illustration of the processes involved, a zone is selected for each image which shows the synchronisation of layer 5 and displays the signals of the synchronisation detector.

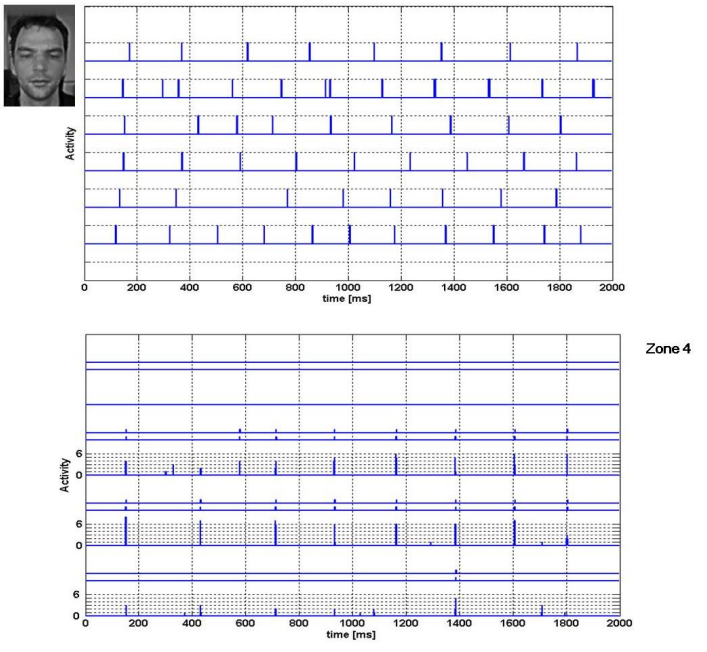


(a) bright illumination

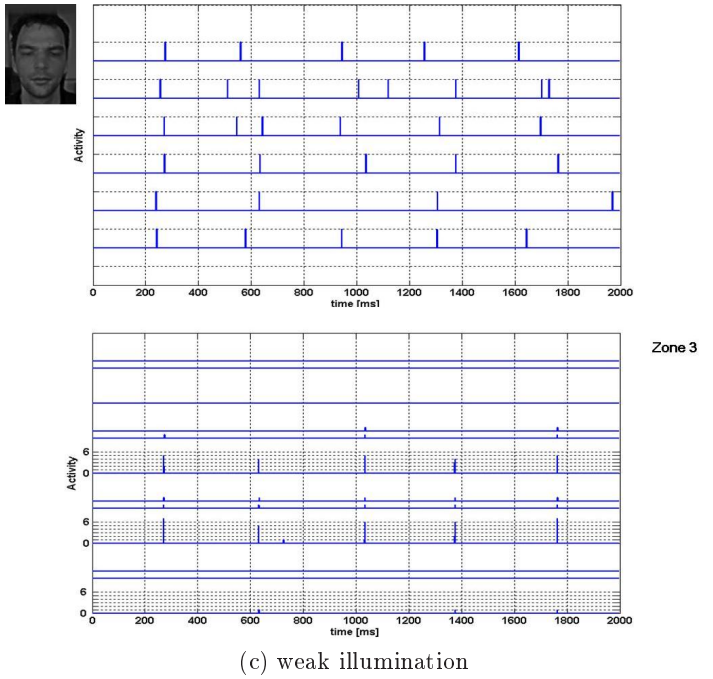
Fig. 7.14 Signal response of the feature recogniser in zones $1-6$

7.7 Invariance of Function as to Form and Mimic

We restrict ourselves in this chapter to a head in frontal view at a fixed distance, and allow for such rotations only for which the object components do not disappear. We will be using the same zones as described above.



(b) mid-bright illumination



(c) weak illumination

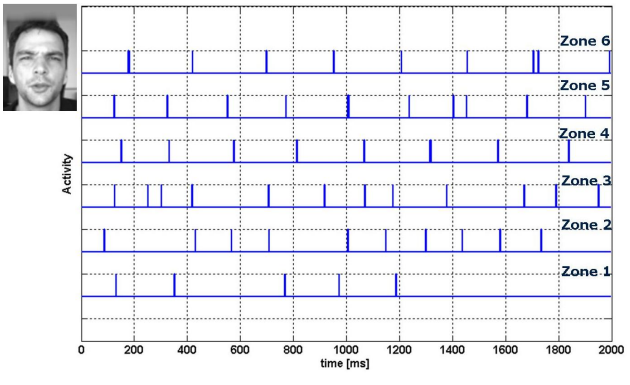
Fig. 7.14 (continued)

In Figure 7.15a a head is shown with four different facial expressions. In consequence, the brows, say, are appearing at different locations. The feature detection is carried out according to Figure 6.12a by means of horizontal Gabor wavelets (one for each zone of an object component). After that, layers $3-5$ (the latter including the enhanced synchronisation detectors) follow, and the summation of all rows of layer 5 by means of neuron N_5.

(a)

Fig. 7.15 Reference head, with 3 further facial expressions

Thereafter we test the proposed architecture with the different facial expressions. The results are shown in Figure 7.15b-e. For each zone N_5's response for the respective feature is recorded as a function of time.

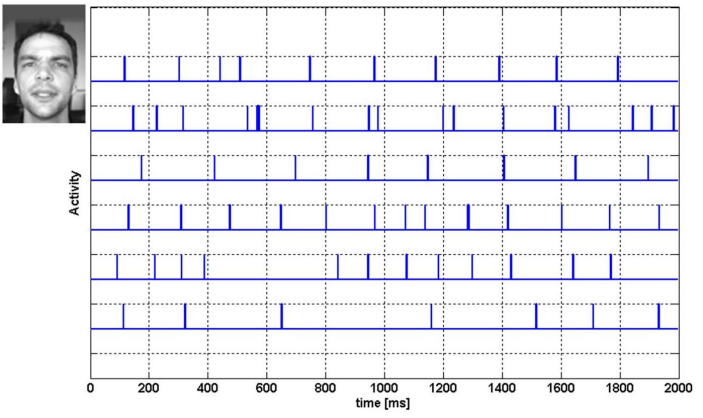


(b) Reference mimic

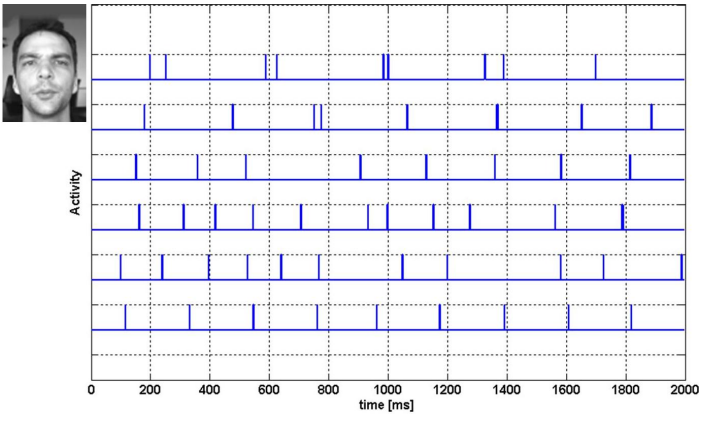
Fig. 7.15 Signal response of the feature recogniser for zones $1-6$

The pulse activity begins to stabilise for all zones after about $100\,ms$. In zones of weaker feature intensity, the synchronisation takes longer, but generates unambiguous responses.

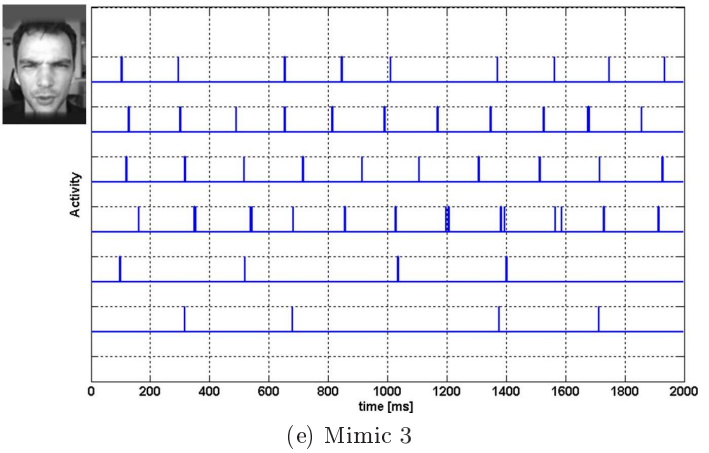
We note in passing that the simulation of the 4 faces on a high-end PC took two full days, without interruption by any other process.



(c) Mimic 1



(d) Mimic 2



(e) Mimic 3

Fig. 7.15 (continued)

7.8 Generating Object Components through Binding of Features

In the last chapter a zone and the object component, resp., could possess a single feature only. Let us now consider the general case of several features encoding a single object component. In the example of the brow-eye section, besides the horizontal and vertical Gabor wavelets, further ones for the detection of the skewness etc. would be needed, for example.

Each feature detector determines the strength of the appearance of a feature. On the one hand, the strength is represented by the pulse rate of neuron N_5 which is proportionate to the illumination level, and on the other hand by the correlation of the pulses in layer 5. The correlation can be negligible, or appear as perfect synchronisation, depending on whether the feature is lacking or clearly extant. From now on we designate neuron N_5 as the *recognition neuron*.

Assuming n features per zone, the concatenation of the signals S_1 to S_n produced by neurons N_5 of the feature recognisers must be realised independently from the temporal sequence of firing of the neurons N_5 and their pulse rates. It is exactly the existence of pulse activity of neurons N_5 that informs us about the presence of all features. Note that the time ordering of the signals S_1 to S_n - assuming all features stay present for a while - changes with time because the pulse rates of the neurons N_5 differ generally. Hence, a net is sought that detects the coexistence of the signals S_1 to S_n.

The net shown in Figure 7.16 achieves exactly this. Its mode of operation draws on the one of the „2-before-1“ detector which constitutes the first part of the pulse difference detector (chapter 5). Let us inspect the first row in Figure 7.16 and start with the left two neurons. Either S_2 preloads and S_1 makes the neuron fire, or the converse must happen. In this way we can detect whether S_1 fires before or after S_2. Moving to the row further to the right, there are successive tests as to whether S_3 fires before or after the pair (S_1,S_2) and whether S_4 is ahead or after S_3 if S_3 fires before or after the pair (S_1,S_2). 8 of the 24 possible temporal sequences of the input signals S_1 to S_4 can be detected in the first row in this way. In the second row, it is a further 4, in the third row another 6, and in the next three rows further 2 sequences each. Thus, if every neuron N_5 fires at least once, the neuron N_Z fires immediately.

The first row is already sufficient to indicate unequivocally that the object component is extant. This is because one of the sequences detectable in the first row will appear eventually, provided all features are present. Accordingly, the zone neuron N_Z will fire. Thus the other rows speed up the detection of the zone. From Figure 7.16 it follows also that neuron N_Z can not fire unless all features are present.

Since any change of an image causes a portion of the neurons to be preloaded, a faulty firing of the zone detector neuron may happen. The simple circuit shown in Figure 7.17 suppresses such a behaviour. If neuron N_R transmits a pulse, neurons N_1 and N_2 will also fire, irrespective of how much

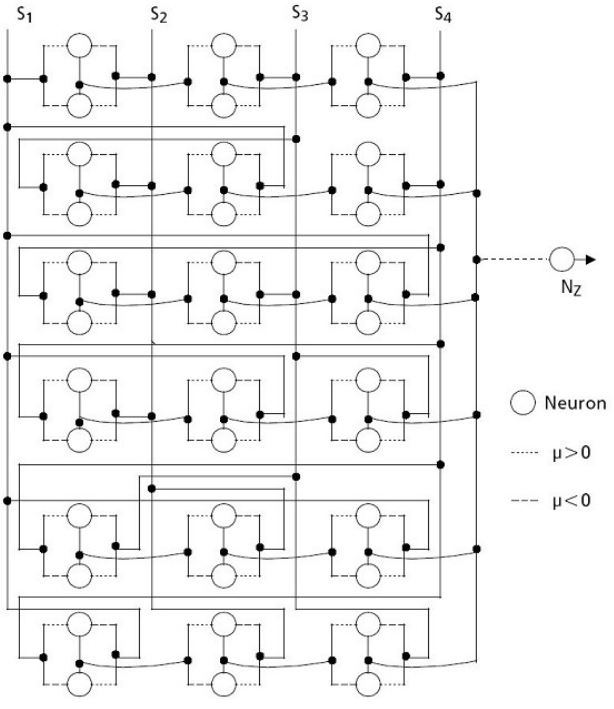


Fig. 7.16 Zone detector's architecture with basic modules

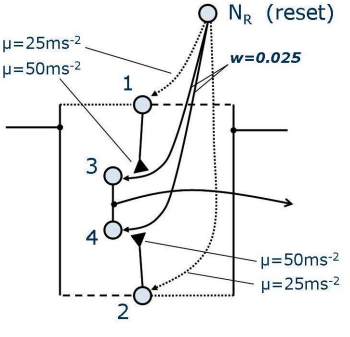


Fig. 7.17 Zone detector module with reset circuit

the membranes are preloaded. The neurons N_3 and N_4 do not fire, however, because with the choice of the positive sign (chapter 5, section 5.5) their dendritic synapses function as anti-coincidence detectors. This way a reliable discharging of the neurons N_1, N_2 is accomplished. Conversely, if an input pulse is superimposed on to a reset pulse or if no reset pulse exists, N_3 and

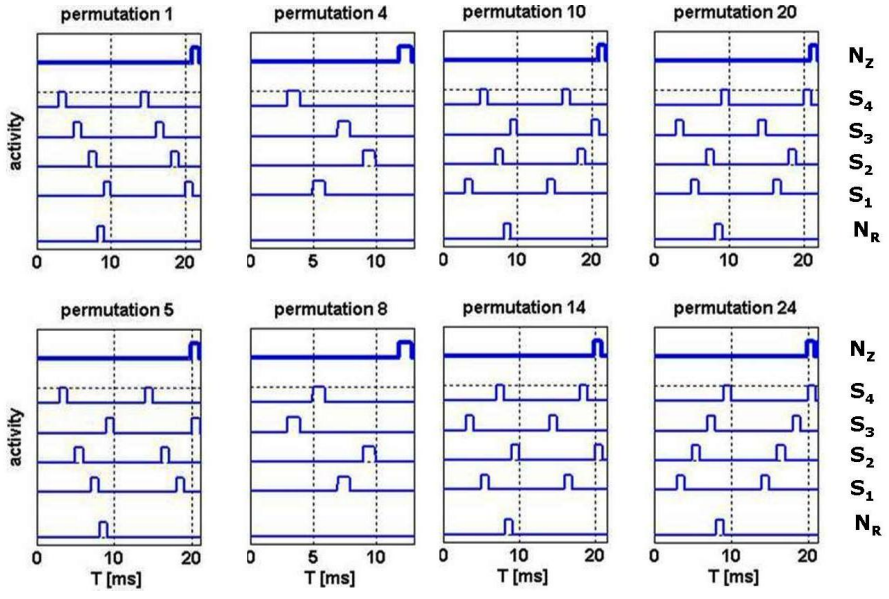


Fig. 7.18 Signal behaviour of the zone detector

N_4, resp., generate pulses as in Figure 7.16. Obviously, the reset frequency must be chosen higher than the highest input pulse rate.

Because it is possible to derive the behaviour of the zone detector through analysis, an experimental confirmation is not necessary. Nevertheless, for the sake of completeness, the firing of the zone detector is displayed in Figure 7.18 as a function of the concurrent input signals $S_1 \ldots S_4$ and the reset pulse of N_R, for some of the 24 possible combinations of the input signals. It becomes clear that a 4-group of signals only let the zone detector fire if there is no reset pulse present.

The extension of the architecture is obvious for the case of an object component, the existence of which should be detected when some, but not all features, exist. Furthermore, the pulse rates of the synchronisation detectors, ie. the respective brightnesses might be helpful in detecting an object component.

7.9 Summary

Robust object recognition begins with the robust recognition of object components. The feature recogniser consists of a plane net which is connected by constant and equal synapses in 1-to-1 fashion with the detectors of a special feature type within a zone. The interconnection of the neurons of the recogniser is never fully connected or of the simple closest adjoining type, but needs

to be optimised. With the presence of a feature, those neurons which lie above the responding detector neurons will fire synchronously. The strength of synchronisation is measured with the synchronisation detector, and dependent on the application, the minimal number of neurons can be configured that must fire synchronously before a feature is classified as extant. The feature recognition we presented functions quite independently of the exact location of the feature within its zone, and it is sufficiently robust to cope with any distortions the feature can make. A further inherent property is the independence from variations of the brightness of the object.

In case of an object component comprised of several feature types, the outputs of the respective feature recognisers are processed by the zone detector. Its function is simply to detect the co-existence of the signals of the various recognisers independently of their individual pulse rates. The zone detector uses dynamic synapses and essentially counts the feature types. If and only if all (or any fixed number of) features exist does it generate a pulse. Thus, the function of the zone detector is the binding of various features to form an object component, independent of pulse rate, deformation and illumination.

8

Nets for Robust Head Detection

Ulrich Ramacher and Arne Heittmann

In order to generalise about the available head-specific results for a representative number of heads, two ways seem to be promising: the one determining a minimal number of zones valid for all heads, or the other individually adapting the distances between zones for all heads. The first way is attractive, as the architecture already exists and does not need any extension. However we expect a progressive overlapping of some zones with an increasing number of heads, which may downgrade the quality of recognition compared to the 1-head case. The second way requires a new architecture which we don't have at hand now. Both ways seem to be useful. For instance, the class of chairs does not allow for a description similar to heads. If backrest, seat and chair leg are chosen as object components, each one exists in a great variety of ways. Therefore, a description of fixed zones does not seem to work in this case.

In this chapter the zones are determined for a representative number of 1500 heads (Figure 8.1).

Fig. 8.1 Optimal zones, extracted from 1500 heads

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 141–148, 2010.
© Springer-Verlag Berlin Heidelberg 2010

The zones are concentrated on eyes, brows and mouth and take note of diverse positions of the features, when people have different facial expressions. A horizontal Gabor wavelet is attached to each zone. A point of a zone relates to the N_7 layer of the Gabor wavelet implementation, ie. their receptive field may be situated up to about 3/4 outside of the zone.

The task remains to demonstrate a head detector which is independent of illumination and mimics, and when faces are presented in frontally fixed positions which take up the whole visual field. Since the selection of zones is performed by a statistical method, the feature architecture from the last chapter remains unchanged. The aim is solely to construct a net for object recognition, in this case a head, and this, clearly, must be based upon the available recognition of object components.

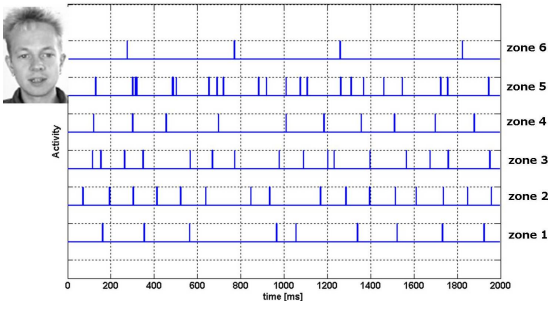
8.1 Results of Head Detection

The existence of the respective object components in the six zones is assumed to be determined by the zone detectors as discussed in the last chapter. The signals $Z_1 \ldots Z_6$ are then input into a net which is constructed according to the instructions in chapter 7, Figure 7.16. The outputs of the single rows are directed to the *object neuron* N_O, and this one will fire, as we know, only when all six zones are present in a head image. Thus the object components are bound together to form an object in exactly the same way as object components were generated from different features. As mentioned in the last chapter, the object detector may be complemented to deal with fewer zones than six. In this book, we restrict ourselves to $n = 6$.

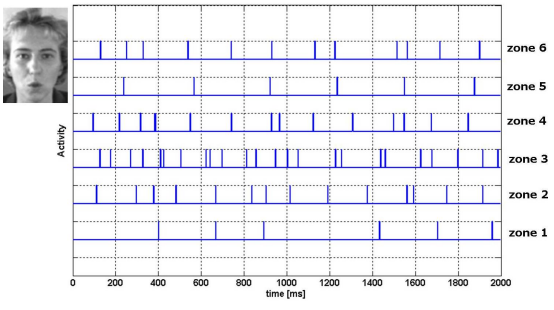
Since there is already proof of the invariance of illumination changes for each zone, a similar invariance can be expected for the object detector. In chapter 7 the zones were particularly chosen for the head under consideration. Now, the zones are extracted from 1500 heads. Hence our task consists of checking for the expression and pose invariance of all heads.

As the simulation of a head with 4 different poses took 2 days, a full test can not be conducted. Instead, such heads were selected from the gallery which would stretch our architecture to the limit: beards, glasses, weak contrasts, different sexes. In Figures 8.2 a-n the activity diagrams of the 6 zones and the object detector are shown for 14 selected heads. The more complex the head images become, the more the number of active zone detectors and the object recognition, resp., decreases.

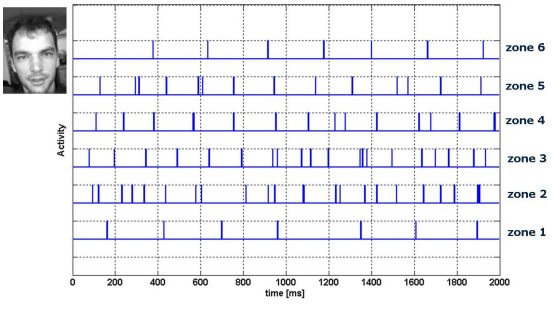
We believe these results prove the usability of the head detector for frontal poses at a fixed distance. Refined adjustments of the feature-specific connection in layer 5 of a feature recogniser, and - more importantly - more features carrying additional information about the whole scene will, in future, allow us to augment the recognition rate effectively - without using more than the recognition chain presented so far.



(a) Head 1, all zone detectors respond easily

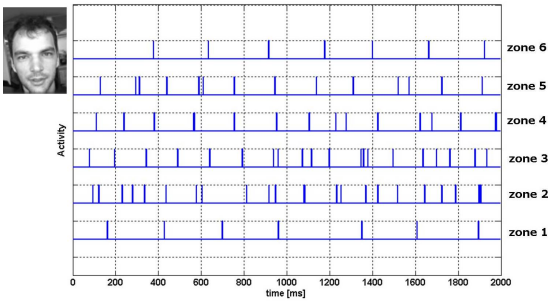


(b) Head 2, dito

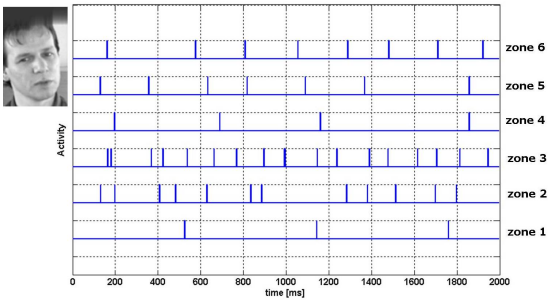


(c) Head 3, dito

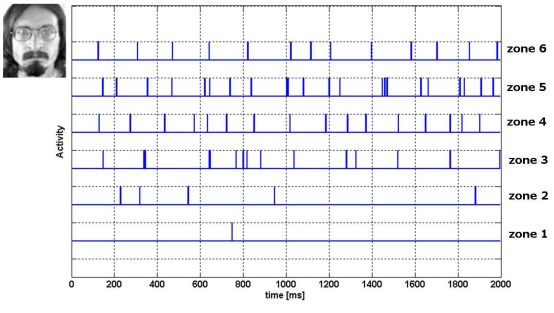
Fig. 8.2 Acivity diagrams of the six zone detectors (heads 1-3)



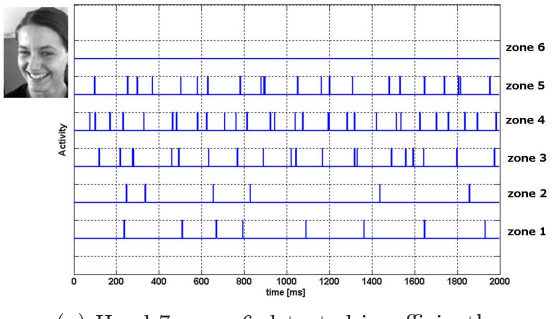
(d) Head 4, dito



(e) Head 5, dito

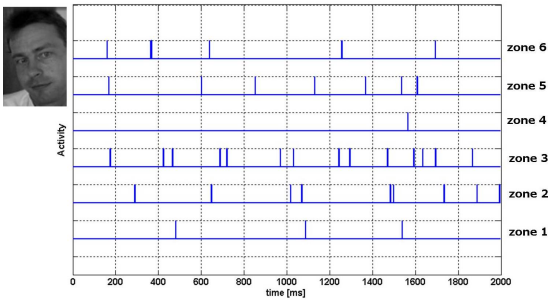


(f) Head 6, zone 1 detected insufficiently

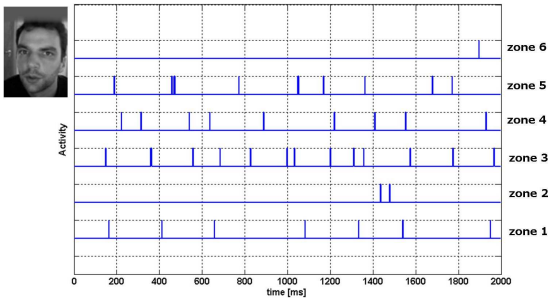


(g) Head 7, zone 6 detected insufficiently

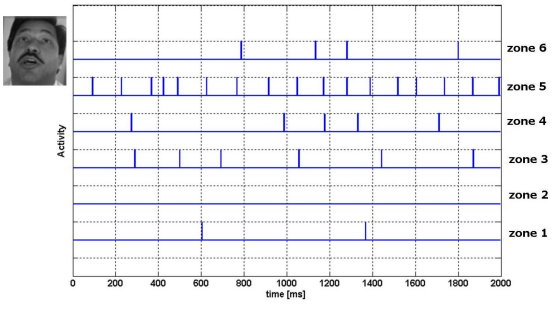
Fig. 8.2 Acivity diagrams of the six zone detectors (heads 4-7)



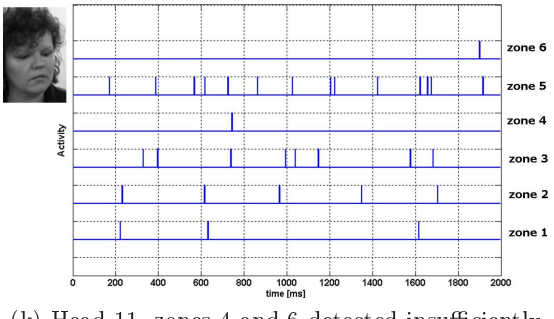
(h) Head 8, zone 4 detected insufficiently



(i) head 9, zones 2 and 6 detected insufficiently

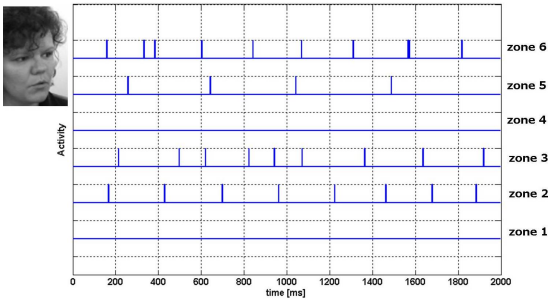


(j) Head 10, zones 1 and 2 detected insufficiently

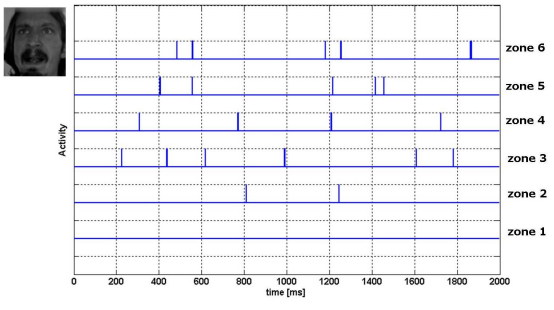


(k) Head 11, zones 4 and 6 detected insufficiently

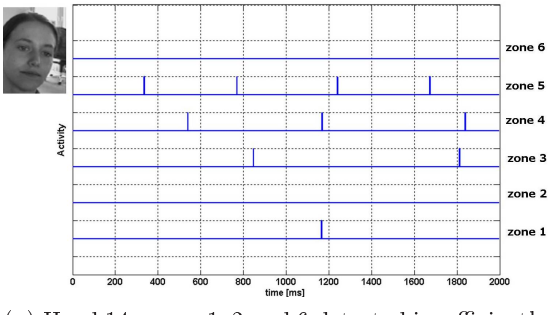
Fig. 8.2 Activity diagrams of the six zone detectors (heads 8-11)



(l) Head 12, zones 1 and 4 detected insufficiently



(m) Head 13, zones 1 and 2 detected insufficiently



(n) Head 14, zones 1, 2 and 6 detected insufficiently

Fig. 8.2 Activity diagrams of the six zone detectors (heads 12-14)

8.2 Next Steps

So far we have drawn upon the frontal view and a viewing distance covering the visual field. In the next step we would have to consider larger rotations and/or partial occlusions of the head or images with far more distant heads, such that some object components would no longer be visible. Thus the head detector presented must be complemented in many respects, or additional head detectors added, the signals of which would be conducted to what is then rightly called a *head recogniser*.

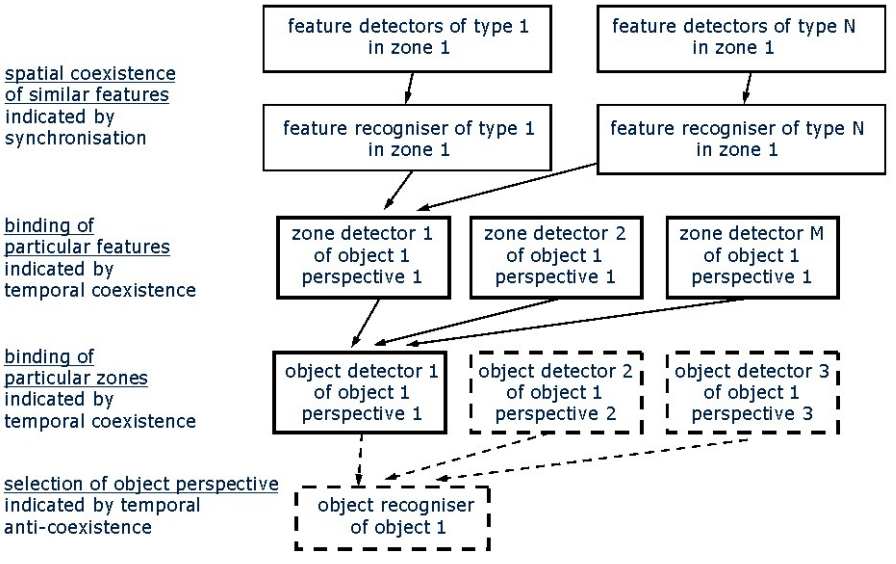


Fig. 8.3 Status of the vision architecture

Let us consider, for instance, heads in profile. Then, the zones would have to be constructed anew, and a new head detector would emerge. The same would hold for heads viewed from the rear. Since only one of the different head detectors should respond, an object recogniser could be constructed as a simple anti-coincidence circuit as in Figure 5.19, but now with $\mu>0$.

Figure 8.3 shows the scheme of the vision architecture developed so far, as much as it was confirmed by experiments and it appears to be expandable according to the considerations given above.

Next we consider heads partially covered by other objects. For instance, an inclined head with the eyes covered by a hat. In this case mouth and nose zones may still contain sufficient information for head detection. This will certainly not be the case when, in addition, a shawl hides the chin and mouth. Only if the human figure is known as an object built from object components like a head etc. and relating to other objects like a hat etc. would it become possible to detect the head.

In consequence, a head recogniser must check for the information of other related objects at its input side. Just as a feature recogniser senses the feature's world of appearances - thereby creating the necessary robustness - the head recogniser is thought to perform a similar function. Nevertheless, the architecture of the head recogniser must be different from the one for the feature recogniser. First of all, it is no longer necessarily a head detector that responds, but may be about a quarter of all the zone detectors of a head. Hence, the scene must be investigated that head is part of, and the existence of several other objects must be detected that interact with the head. Imagine,

for instance, a swimming person, the head covered with a huge sombrero such that no profile or part of the head is visible. In consequence, it seems more adequate to speak of a „head concluder“ than of a head recogniser.

For such a scene analysis, a prerequisite is the existence of a correlation matrix in which the various objects relating to a head, say, are connected similarly to the scheme ruling an object and its object components, presented in the last chapter. Compared by function and not by implementation, we expect such a correlation matrix to resemble what is known as associative memory.

A further prerequisite for a head recogniser is the property of the feature cascade to detect features at various distances (distance invariance). Depending on the distance, the zones will turn out to be smaller than in the reference position, and the distances between zones will vary accordingly.

Furthermore, the decomposition of the retina image into Gabor wavelet responses precedes any further processing and, hence, touches on the central question of how the brain accomplishes, for instance, the seizing of the stem of an apple which requires a pixel-exact positioning of the fingers.

The answers to the last two questions are given in the next chapter.

8.3 Summary

The architecture of vision presented so far is to be considered as a first step only, even though important results were obtained in the former chapters. The mechanism of feature detection, the binding of features to object components and object components to objects independent from the level of illumination, are universally available. Furthermore, the problem of the multiplicity of appearance of object components is solved by the zone architecture according to Figure 7.6 and 7.8. The next steps must be aimed at the distant invariant object recognition and the construction of an object library which is not only comprised of heads, but also of cars, tables, flowers, etc.

This will give rise to considerable trouble, for zones and feature types have to be defined for each new object. It is expected that chairs, for instance, will need to be described by far more complex features than those of heads. Nevertheless, the requirement will always be to bind features to object components and object components to objects in a robust fashion. And invariance will always have to be realised as to illumination, distortions, rotations and distance (for the latter, see the next chapter).

Taking into consideration the multitude of effects developed by the simple neuron and synapse model in the course of building basic information processing structures, it appears that the construction kit is sufficiently filled to start the design of an object library.

9

Extensions of the Vision Architecture

Arne Heittmann und Ulrich Ramacher

In the last chapter, we have raised the following questions:

1. How must the present architecture be extended if distance and scale invariance, respectively, has to be supported?
2. How can it be possible that we perceive the world as time-continuous images, although we are using a pulse presentation for the whole recognition chain and the retinal image is decomposed into features in the very first processing step?
3. How can objects with an almost unlimited diversity of appearances (e.g. chairs, plants, ...) be recognised?

We are able to give a contructive answer with respect to the first two questions in this chapter; for the last question see the next chapter.

9.1 Distance-Invariant Feature Pyramid

First of all it is essential to identify the subsystems of the human eye which are relevant for feature extraction which is invariant to distance. The fundamental parameters are:

(a) optical resolution (pixels/mm^2),
(b) the region of visual field which is used for object recognition.

In general, photo receptors are classified into so-called *rod cells* and *cones*. Depending on the species, 2-4 different types of cones with different spectral susceptibility are found in vertebrates and these enable the perception of colour. On the human retina three types of cones are integrated: (a) the so-called L-type (or receptor for red), (b) the so-called S-type (or receptor for blue) and (c) the M-type (or the receptor for green). At sufficient light intensity cones become active. Besides the discrimination of colour, cones play an essential role for the detection of optical contrast and spatial resolution.

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 149–171, 2010.
© Springer-Verlag Berlin Heidelberg 2010

The susceptibility of cones to light is low compared to rod cells. Cones typically operate both under daylight conditions and under gloaming conditions. As soon as twilight or darkness comes, only rod cells are sensitive to the residual light. Due to their high susceptibility to light, rod cells saturate already at low light intensities.

Both cones as well as rod cells are linked to so-called ganglion cells, which in turn are linked to the visual cortex via the optic nerve. The ganglion cells are commonly seen as a gateway between the eye and the visual cortex. The enhanced susceptibility of rod cells is not only due to an enhanced receptor sensitivity, but also a consequence of the underlying network: many rod cells are linked via bipolar cells to a few ganglion cells. The direct consequence is a reduced spatial resolution due to the averaging properties of the network. The pixel information received from these rod cells is not suitable for the detection of local contrast.

However, individual cones are directly linked to individual ganglion cells without a widespread averaging network. Here, a precise localisation of a light stimulus is possible, even for the subsequent processing chain in the visual cortex. In summary, we can reasonably assume that the information used for detection of contrast originates from the cones.

In the following, only those properties of the eye will be considered which are definitely needed for the detection of static spatial structures. Since the rod cells are mainly used for scotopic vision with poor spatial resolution, as well as for the detection of motion at the boundary of the visual field, the influence of rod cells on the detection of contrast will be disregarded. Despite a total of 120 million rod cells, those cells contribute only marginally to the detection of edges or lines. For this particular purpose cones are favoured, since they provide a sufficiently high spatial resolution. With a total of 6 million cells, mainly cones contribute to photopic vision, which is the focus of the following considerations.

Apart from functional differences, retinal cells are distributed over the retina with diverse densities. The highest density of cells is found inside the so-called *macula lutea* (yellow spot). From the quality point of view, the spatial resolution as well as perception of colour is outstanding inside this area. Also, within the macula lutea along the optical axis of the eye the so-called *fovea centralis* is located. Here only cones but not rod cells are found. The density of cones decreases from the centre of the *Fovea Centralis* towards the boundary of the retina.

The density of rod cells increases beyond the boundary of the *Fovea Centralis*. When an excentricity of 20° (solid angle oriented to the optical axis) has been reached the density of rod cells decreases towards the boundary of the retina (Figure 9.1). It was experimentally found that areas near the boundary of the retina have a higher susceptibility to motion, and it is commonly assumed that the rod cells located near the boundary of the retina are responsible for the detection of motion. It is further assumed that reflexes are derived from the response of these motion detectors in order to turn the eyes

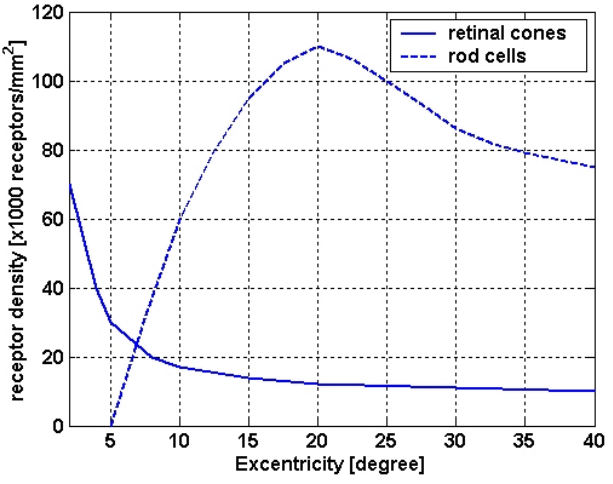


Fig. 9.1 Overall distribution of L-, M-, S-cones, and rod cells on the retina

in the direction of the interesting scenes. As soon as the eye is focused on the scene, cones are used to analyse the details.

The cones representing the L signals as well as the M signals and the S signals are not directly connected with the ganglion cells. Instead, three equivalent signals are generated at the retina by means of special neural circuits (Figure 9.2):

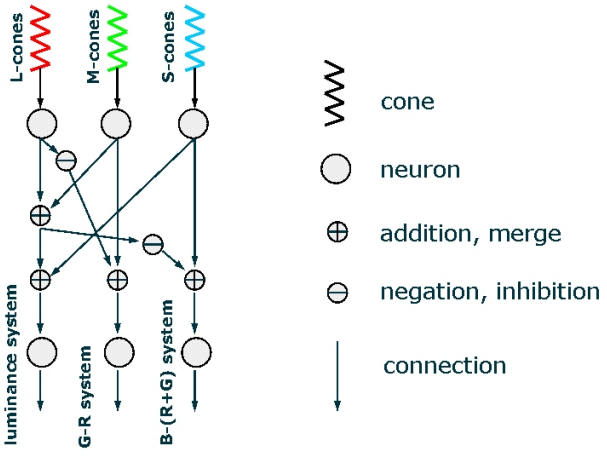


Fig. 9.2 Local wiring of cones and generation of (a) luminance signal, (b) G-R signal and (c) B-(R+G) signal.

(a) the luminance signal (R+G+B),
(b) the red-green signal (G-R) and
(c) the blue-yellow signal (B-R-G).

It is possible to produce a detector sensitive to edges and lines and which depends only on the luminance signal. Since three colour signals are merged into a single luminance signal, the total number of luminance pixels on the retina is calculated to $2{\cdot}10^6$. Furthermore, the number of luminance cells which are located within an excentricity of 20° (equivalent to 40° of the visual field) is limited to $1 \cdot 10^6$ (Figure 9.1).

If the observation distance is kept between 50 cm and 80 cm, in the particular case of detecting a head, the required solid angle can be reduced to 20°. For the following considerations, however, we stick to a solid angle of 40°. This extended angle is sufficient for the detection of arbitrary objects which have to be recognised on different scales, i.e. especially at different distances from the observer. The recognition system (i.e. feature cascade and object recogniser) should be designed in such a way that it is possible to extract and process a sufficient amount of features on every scale and from every perspective. With regard to the feature cascade, this implies a sufficiently large number of scales (i.e. spatial frequencies) and a sufficiently large number of orientations on each scale.

In order to estimate an upper bound for the system complexity several parameters have to be taken into account:

(i) the number of layers for different resolutions (scales),
(ii) the number of orientations on each scale,
(iii) the locations on the retina where features have to be detected.

We let the parameterisation be governed by the requirement that the set of feature detectors should allow an almost complete extraction of (static) information contained in the image. The parameters of the required scales (i.e. the stepping of spatial frequencies) result from considerations towards scale invariant object recognition. At a predefined distance between object and observer, the object is projected onto the retina. At this particular distance all characteristic features can be extracted using an optimised set (concerning spatial frequency and orientation) of detectors. If the distance is increased, the projection of the object on the retina is seemingly scaled down. In particular, distances between adjacent edges and adjacent lines appear to be smaller. It follows that the receptive field of a feature detector covers an area which is larger than the area covered by the downsized pattern. Consequently, the detector responses are determined by additional pixels surrounding the specific features we are actually interested in. If the distance of the object increases further, the number of surrounding pixels which contribute to the detector responses dominate. Hence, they can lessen or even completely destroy the

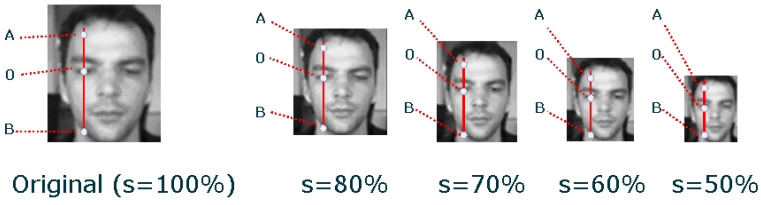


Fig. 9.3 Head pose observed at different distances. The parameter s represents the visual factor of scaling.

response. In order to exemplify the effect outlined above, a head pose given in Figure 9.3 was examined experimentally. The factor s specifies the relative horizontal size as well as the relative vertical size of the object compared to a reference distance. $s = 1$ specifies the object in reference distance. For $s = 0.5$ the object is downsized by a factor of two, so it is positioned further away. It is possible to design the set of detector parameters in an optimal way for objects positioned in reference distance to the observer. Subsequently, such a set of optimised detectors is taken and reused when objects are positioned far away. In Figure 9.4 there are illustrations of the activity distributions of detectors

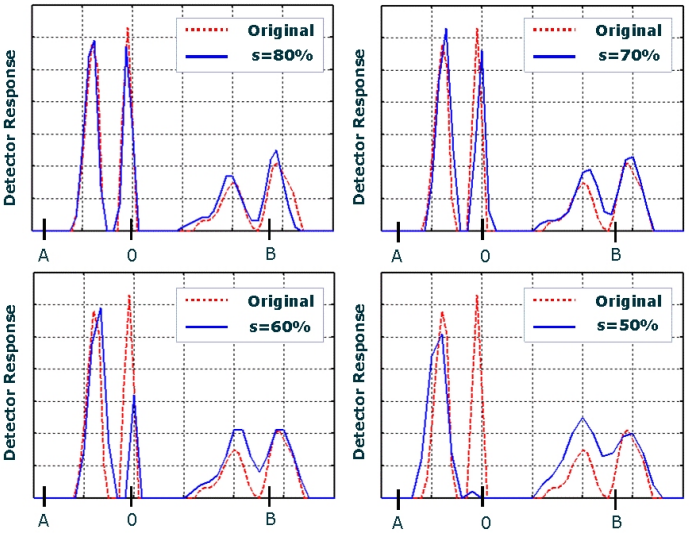


Fig. 9.4 Activity distributions of horizontally orientated detectors for different values of the parameter s.

which are sensitive to horizontally orientated edges. The detectors are located on the intersection $A - B$. The feature located near the position 0 is robustly detected up until $s = 70\,\%$ and is completely annihilated at $s = 50\,\%$. Thus this feature is not visible to the object recogniser when $s = 50\,\%$. Likewise the effect of superimposing is observed at location B. Whereas both features are kept apart till $s = 70\,\%$ the ability to clearly distinguish both features degrades from $s = 60\,\%$. Both features seemingly fuse into a single feature at $s = 50\,\%$ although both edges still are resolved on pixel level.

In order to detect the scaled-down features, the receptive field has to be scaled down and the spatial freqency to be scaled up. Hence, additional detectors with different parameteres are required. Let us consider different resolution scales, from the highest resolution for objects which are far away, towards modest resolutions for objects which are in reference position (Figure 9.5, layer A-E).

The spatial frequency for detectors on every next scale has to be halved at $s = 50\,\%$. The dimension of the receptive field appears to be doubled in the direction perpendicular to the wavelet orientation, since the complete cosine wave with halved frequency has to fit in the extended receptive field. Likewise, it is necessary to lengthen the wavelet along the wavelet axis in order to keep up the sensitivity with respect to orientation. The overall area of the new receptive field is four times larger compared to the size of the reference wavelet. For reasons of economy we keep the number of sampling points identical for any wavelet on any scale. Here the size of a sampling point on the retina

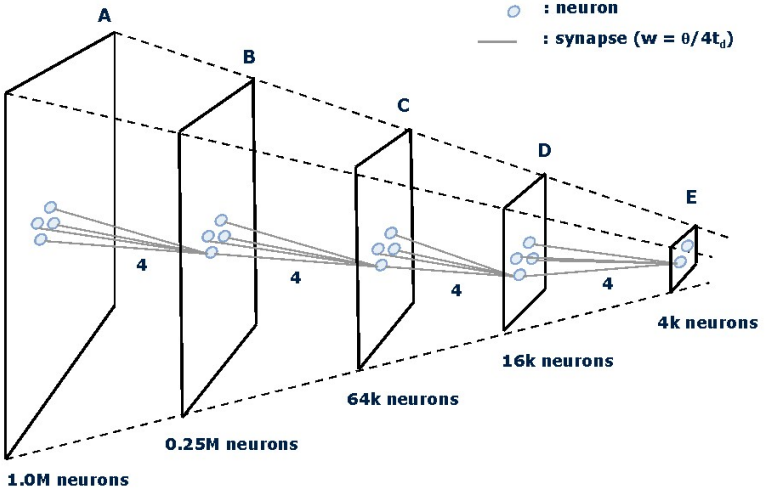


Fig. 9.5 Resolution pyramid for $s = 50\,\%$. In layer A neurons are located which encode the luminance information from the cones. The image information is scaled down on each additional layer.

is enlarged by a factor of four by averaging over four pixels. The averaged signal is converted into a pulse signal and represents the input signal for the particular scale. The spatial arrangement of the detectors also changes with the enlargement of the receptive field. In particular, the horizontal distance and the vertical distance between adjacent detectors are doubled. Therefore the total number of detectors located on each scale reduces by a factor of four from scale to scale.

The overall number of different orientations roughly matches the number of different wavelets which can be realised on a particular scale with a modest overlap. Let us consider wavelets selected as an example with a receptive field of 7×7 pixels. If we stick to this size, wavelets with 4 and 8 different orientations can reasonably be discriminated. If we arrange wavelets of identical orientation and scale which cover the whole retinal area, a pairwise overlap of profiles along the wavelet axis results in a redundant extraction of information. Again, it is reasonable to widen the sampling grid along the wavelet axis for reasons of economy. Along the wavelet axis we take a 2-pixel grid, and perpendicular to the wavelet axis we take a 1-pixel grid. If a particular scale has $n \times m$ pixel, about $n \times m/2$ detectors (per orientation) are arranged in a 2-dimensional way.

At the transition from one scale to the next, pixel information is averaged, which can be seen as the formation of an image with a (desired) lower resolution. Receptive fields of size 7×7 pixels can be defined on the new scale, which in turn prepares the input information for particular feature detectors. This procedure is repeated for each scale of the feature cascade. Thus a corresponding image layer emerges for each additional wavelet scale. The pixel information on layer $i+1$ can be derived from the pixel information on layer i. The formation of any pixel on layer $i+1$ is given by the averaging of $1/s^2$ pixel of layer i as defined by the parameter s. A resolution pyramid for $s = 50\,\%$ is illustrated in Figure 9.5. Here 2×2 pixels are averaged on each layer to form new pixels on the next layer.

In the following we will demonstrate that the pyramidal architecture is well suited for the implementation of wavelets with different spatial frequencies. Let us consider one particular wavelet on scale C, and another particular one on scale D. Both wavelets are linked via a receptive field of 7×7 pixels to the respective pixel layer. The receptive fields on scale C have a size of 28×28 pixels with respect to layer A, while the receptive fields on scale D have a size of 56×56 pixels (also with respect to layer A).

In order to examine the detector profiles, samples are taken on the receptive field, using spots to be exposed on the retina (Figure 9.6). Spots with a size of 8×8 pixels are used for sampling the profile of the wavelet located in layer D, while the spots for sampling the wavelet located in layer C have a size of 4×4 pixel. The sampling is carried out either by means of bright spots with a dark background (*Bright Spot*) or by means of dark spots with a bright background (*Dark Spot*), see also chapter 6.

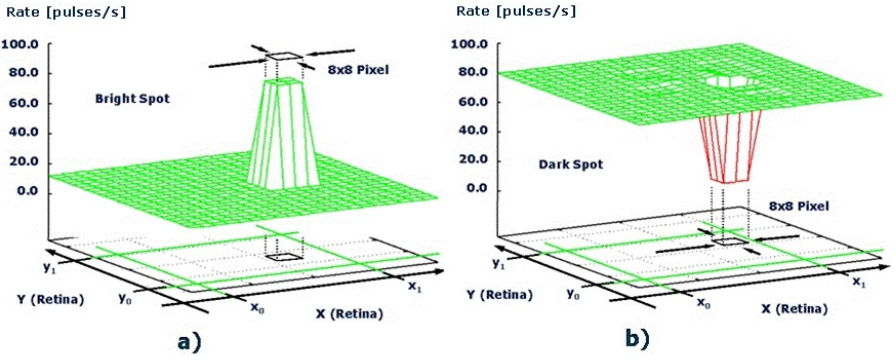


Fig. 9.6 Pulse rates of particular neurons located in layer A for a given stimulus. a) *Bright Spot* and b) *Dark Spot*.

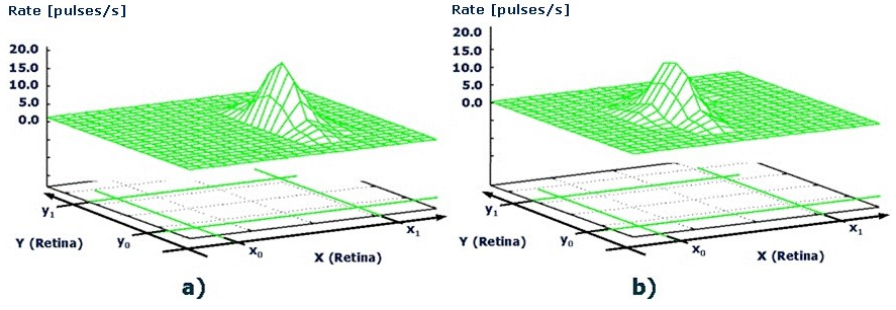


Fig. 9.7 Response of a detector which is located in layer D, dependent on coordiantes of layer A (luminance signal of the retina)

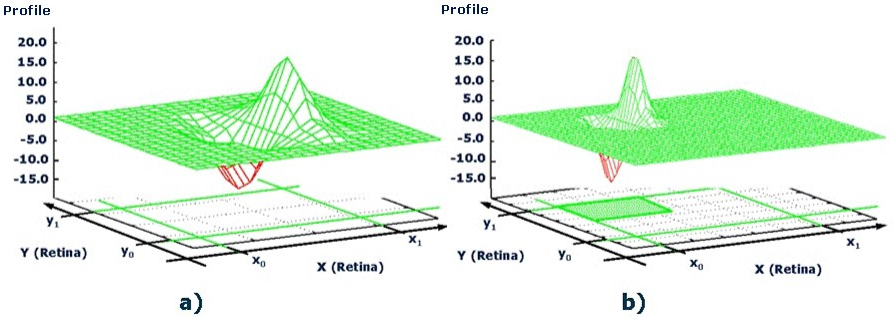


Fig. 9.8 a) Detector profile of a feature detector which is located in layer D. b) Detector profile of a feature detector which is located in layer C. The receptive field covers 25 % of the receptive field given in a).

The measured partial detector profiles are illustrated in Figure 9.7. The characteristic half-waves can be clearly seen. The final detector profile is obtained by pixel-wise subtraction of partial detector responses (cf. Chapter 6). Results for layer C as well as for layer D are shown in Figure 9.8. The profile of layer D shows halved spatial frequency (as expected) compared to the profile of layer C.

Area Saving Implementation of a Feature Detector

No matter what orientation it has, each feature detector is linked to the appropriate pixel layer via a receptive field of size 7×7 pixels. Typically, only a subset of sample points inside the receptive field essentially contributes to the detector response. Experimental results have shown that it is enough to consider only 20 sampling points within a receptive field of 49 pixels to sufficiently represent particular wavelets. Examples of distributions of sampling points for 6 different orientations are illustrated in Figure 9.9. As before, we assume that detectors are realised with 64 gradient detectors, i.e. 128 inputs linked to the gradient detectors have to be distributed to 20 sampling points on the receptive field. Consequently, some sample points will be scanned repeatedly up to 16 times. By exploiting the symmetrical properties of the proposed wavelet profiles, it is possible to prune the set of gradient detectors. If a particular pair of pixels is linked to n gradient detectors with identical polarity, all these detectors can be replaced by a single gradient detector whose output is weighted by n (Figure 9.10). The equally weighted connections to neuron N_5 (Figure 6.2a) are now replaced by individually-weighted connections. Each connection is proportionally weighted, with the multiplicity represented by the corresponding gradient detector. For the proposed wavelet profiles it

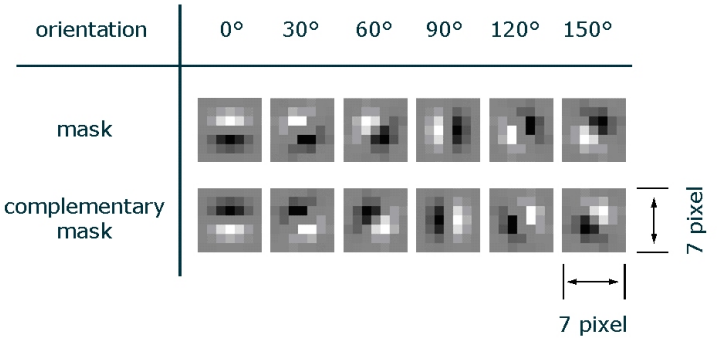


Fig. 9.9 Set of wavelets with a receptive field of 7×7 pixels for 6 orientations and 20 sampling points

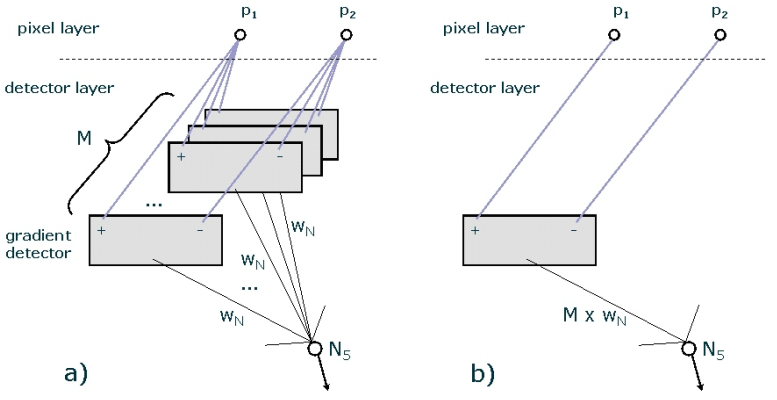


Fig. 9.10 a) Multiple sampling of a pixel pair p_1 and p_2 using M gradient detectors. The uniformally weighted summation is realised using the weight W_N b) Realization using a single gradient detector and weighted summation $M \times w_N$.

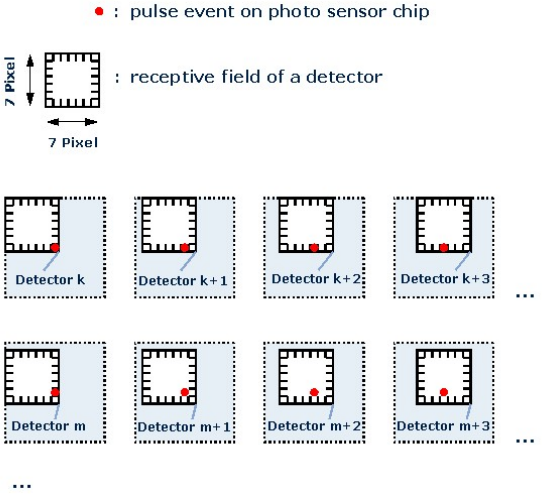


Fig. 9.11 Distribution of a single pulse event to several receptive fields

turned out to be sufficient to connect 10 gradient detectors to 20 sampling points.

To simplify matters, let us now assume that the retina is scanned on a 1×1 grid. If we have a retina with 1024×1024 pixels, we obtain 1024×1024 individual detectors per orientation. Each detector receives pulse events which are generated by neurons located inside the associated receptive field.

If we take a receptive field of 7×7 pixels as a basis, each neuron in the retina is connected to 49 different receptive fields (Figure 9.11). A particular pulse event will activate only those detectors (the associated neurons N_1, N_2) which are connected to a sampling point, i.e. a particular pulse on the retina activates exactly 20 detectors (per scale and per orientation). In addition, a particular pulse affects exactly two gradient detectors per feature detector (Figure 5.11 and Figure 6.2).

Total Effort for the Feature Cascade Including the Object Recogniser

Based on our experiments and these estimations, we can set limits for the effort of realising a feature cascade if we assume a uniform receptive field of 7×7 pixels, as well as 2×64 gradient detectors for each feature detector. For this purpose we assume a particular arrangement of wavelets including a certain grid as was stated earlier: along the wavelet axis we take a 2-pixel grid, and perpendicular to the wavelet axis we take a 1-pixel grid for the placement of wavelets. Estimations for the effort are given in Table 9.1.

	pixels	orientation × grid pattern	neurons per wavelet	synapses per wavelet	Overall number (in 10^6) of neurons/synapses
A	1024×1024	$8 \times 0.5 \times 1$	$(64 \times 2+1) \times 2+4$	$(64 \times 5) \times 2 + 8$	1098/2718
B	512×512	do.	do.	do.	274/680
C	256×256	do.	do.	do.	68/170
D	128×128	do.	do.	do.	17/42
E	64×64	do.	do.	do.	4/10
					Total: 1461/3620

Table 9.1 Estimated number of neurons and synapses based on a symmetric feature cascade with 5 scales, 8 orientations per scale, and a receptive field of 7×7 pixels per wavelet. Each wavelet includes 64 gradient detectors.

If we estimate the human brain to have about 10^{12} neurons and 10^{15} synapses, the feature cascade represents only 0.14 % of the total number of neurons of the brain. The relative proportion of synapses is even lower, barely $3.6 \cdot 10^{-6}$. This picture does not change scale if we take all available luminance pixels (i.e. 2 million pixels) into account.

Let us recall that due to the localisation of features defined by zones a given head pose is specified by the pulse response of only 745 detectors. If we extend the recognition architecture to 5 scales the overall effort is quintupled, provided the zone architecture is replicated on each scale. Then, only 3725 detectors have to be realised in total. However, since we are interested in the recognition of several thousand objects, zones are distributed over all scales as well as over all locations within the feature pyramid.

	zone 1	zone 2	zone 3	zone 4	zone 5	zone 6	total
neurons (10^3)	0.35	0.43	0.46	0.37	0.64	0.39	2.64
synapses (10^3)	2.23	2.72	2.94	2.31	4.19	2.47	16.86

Table 9.2 Additional effort for the realisation of a feature recogniser for different zones. A single scale is assumed.

In addition, the effort for the feature recogniser in each zone must be taken into account (Table 9.2). The additional effort for the zone-base feature recogniser can be neglected if 1000 objects are to recognised. Only in case of 10000 objects comprising 100 zones per object or 100000 objects comprising 10 zones per object the feature recognition architecture needs as much neurons as the architecture for feature detection (Figure 9.12).

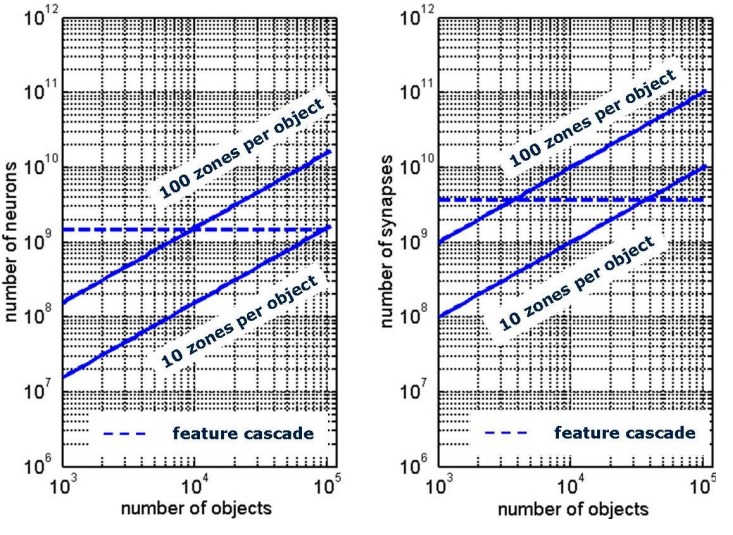


Fig. 9.12 Effort for feature recognition vs. feature detection (dashed line)

Thus, zones are inexpensive to implement and to be utilized accordingly.

Unfortunately, we are only able to speculate about an architecture of a feature pyramid with which to recognise objects independent from their distance. For providing evidence of the expected functionality by simulation is at present hardly possible; for instance, the simulation of experiments with respect to the head detector (1 resolution, 1 horizontal Gabor wavelet) took $1\frac{1}{2}$ weeks on a $2.4\,GHz$-PC containing $2\,GB$ main memory.

On the other hand, we are just reusing architectural elements presented so far - only more types of features as well as more resolutions layers were added.

9.2 The Inner Screen

Let us briefly recall what information generated by the feature cascade is available. An image projected on the retina is expressed on every resolution scale by a two-dimensional grid of vectors, with the number of components of a particular vector matching the number of different features present on a resolution scale. The magnitude of a particular vector component encodes the feature intensity at the respective grid position. Since feature detectors for textures or contours, for instance, occupy an area of at least $2 \cdot 4$ pixels, and the localisation of a contour or a cusp is the result of the cooperation of several detectors, a pixel-like spatial resolution cannot be achieved on that basis without further processing. Furthermore, executing actions like grasping or targeting something demands precise control at the pixel level so that an additional process building upon the responses of feature detectors only is required to recover this information.

A further argument to justify an „inner screen", which embodies a 2D-reconstruction of the original two-dimensional intensity image of the retina, is our ability to recall a picture such as a house from memory before our „inner" eye while keeping our eyes closed. Therefore, we will demonstrate - both mathematically as well as by using a neural architecture - how an image can be reconstructed closely to the original retinal image.

As already explained in chapters 6-9, our approach is based on Gabor wavelets comprising different scales as well as orientations for the extraction of features. A specific wavelet is defined by a so-called wavelet profile g_j(x,y), which is a function of coordinates (x,y). The receptive field RF_{g_j} defines the set of locations where the profile g_j(x,y) obtains values different from 0. Each receptive field of the used wavelets shows itself to be connected as well as bounded. In order to detect a particular feature at a particular location (x_0, y_0) with a wavelet g_j, the wavelet profile is shifted toward the position (x_0, y_0) (in most cases, the center of the receptive field is defined as the origin, cf. Figure 9.13a) and the local pixel information is averaged after weighting with the profile:

$$\begin{aligned} m_j(x_0, y_0) &= \sum_{(x,y) \in RF_{g_j}} I(x, y) \cdot g_j(x - x_0, y - y_0) \\ &= \sum_{(x,y) \in RF_{g_j}} I(x, y) \cdot g_j^s(x_0 - x, y_0 - y). \end{aligned} \tag{9.1}$$

In (9.1) g^s denotes the mirrored profile: g^s(-x,-y)=g(x,y). Then, the feature layer m_j is obtained via a two-dimensional convolution of the image and the mirrored profile (Figure 9.13b).

In general, a set of different wavelets is required to fulfil the demands concerning both, number of orientations and scales. Feature extraction layers $m_1, m_2, ..., m_n$ are thereby obtained, with an individual layer representing a

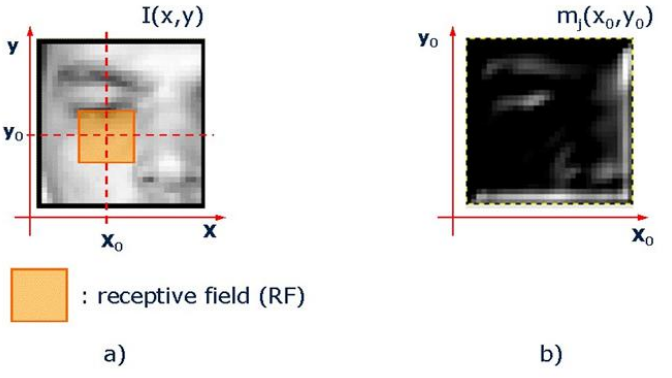


Fig. 9.13 : a) The center of a wavelet is placed at the location (x_0, y_0) in order to extract a local feature within the input image I(x,y) b) Spatial distribution of extracted features for the given wavelet, dependent on (x_0, y_0). The depicted grey values appear proportional to the feature intensity.

particular scale and a particular orientation. Image synthesis will be based on these layers only.

The discrete Fourier transform of g_j^s is defined by:

$$\hat{g^s}_j(k_x, k_y) = \sum_{r_x=0}^{N_x-1} \sum_{r_y=0}^{N_y-1} e^{2\pi i(r_x \cdot k_x/N_x)} \cdot e^{2\pi i(r_y \cdot k_y/N_y)} \cdot g_j^s(r_x, r_y) \ . \tag{9.2}$$

N_x and N_y, respectively, specify the number of pixels along the x-axis and y-axis, respectively. In the Fourier space we obtain:

$$\hat{m}_j = \hat{I} \cdot \hat{g^s}_j \quad respectively \quad \hat{m} = \hat{I} \cdot \hat{g^s}. \tag{9.3}$$

We assume that a local inverse $\hat{\psi}$ of $\hat{g^s}$ exists in the following sense:

$$\forall\ (k_x, k_y)\ :\ <\hat{\psi}\ ,\ \hat{g^s}>(k_x, k_y)\ =\ \sum_{j=1}^{n} (\hat{\psi_j} \cdot\ \hat{g^s}_j)(k_x, k_y) = 1\ . \tag{9.4}$$

Using (9.1) it follows:

$$\forall\ (k_x, k_y)\ :<\hat{\psi}\ ,\ \hat{m}>(k_x, k_y) = \sum_{j=1}^{n} (\hat{I} \cdot \hat{\psi_j} \cdot \hat{g^s}_j)(k_x, k_y)\ =\ \hat{I}(k_x, k_y)\ . \tag{9.5}$$

In the spatial domain this implies:

$$\forall\ (x,y)\ :\ \sum_{j=1}^{n}(\psi_j * m_j)(x,y) = \left(I * \left(\sum_{j=1}^{n}\psi_j * g_j^s\right)\right)(x,y) = I(x,y)\ . \tag{9.6}$$

Because of

$$\sum_{j=1}^{n}\psi_j * m_j = \sum_{j=1}^{n} m_j * \psi_j = I \tag{9.7}$$

the reconstructed image is now obtained by convolving the feature detector layer with the inverse wavelets instead of convolving image I with the Gabor wavelets. Therefore, similar to the functions g_j the functions ψ_j can be imagined as profiles now realised by receptive fields at layers N_5 and N_7, respectively, of the feature detectors. As a direct consequence, the architecture of the feature detectors can be reused almost unchanged if only the profiles of the inverse wavelet vectors ψ are known.

A prerequisite for image synthesis is that the inverse wavelets can be built with localised receptive fields and few sampling points. Given g_j, equations for the Fourier-transformed profiles follow that determine the inverse wavelets by (9.4). But the Fourier transform of a wavelet covers only a small area within the frequency domain. As an example, Figure 9.14 shows the distribution of the spectrum magnitude in the frequency domain for 4 selected Gabor wavelets. The colouring in a) reflects the response which has been scaled to the corresponding maximum value.

Since any individual Gabor wavelet features appreciable magnitudes within a localised frequency range only, a wavelet contributes to the fulfilment of (9.4) in a localised frequency range. Outside of the specific band, the magnitude

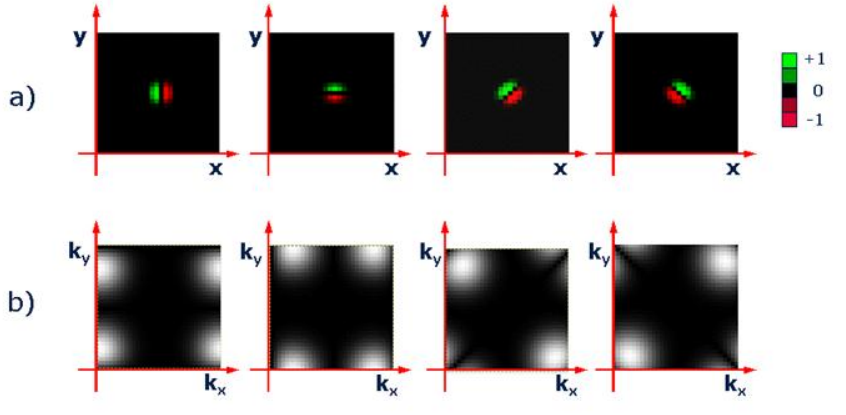


Fig. 9.14 a) Gabor wavelets in the spatial domain, b) magnitude spectrum in the frequency domain.

spectrum of a particular wavelet decreases. Unless no other wavelet features significant values in this outer frequency band, reduced magnitude must be compensated for with reciprocal enhancement of the localised magnitude spectrum of ψ_j to keep (9.4) fulfilled. If there is a major drop in the magnitude spectrum g_j, an ill-conditioned or even irresolvable set of equations emerges. Thus, only stable solutions of (9.4) are of interest, and they must also lead to functions ψ_j with a local receptive field.

It turns out to be very helpful to claim that the set of g_j covers the frequency range in a uniform way, i.e.

$$\epsilon_1 < \sum_{r_x=0}^{N_x-1} \sum_{r_y=0}^{N_y-1} \sum_{j=1}^{n} |\hat{g}_j(k_x, k_y)|^2 < \epsilon_2 \ , \tag{9.8}$$

Both ϵ_1 and ϵ_2, should be as close to 1 as possible. Then, for each point in the frequency domain, at least one wavelet exists which contributes sufficiently to the spectrum of magnitudes such that a local ψ_j can be determined, hence (9.4) is met in the frequency domain. In Figure 9.15 a set of wavelets obtained numerically and its inverse is shown. The colouring reflects the response scaled with respect to the maximum value. Red labels represent negative values while positive values are labeled green. Black represents 0.

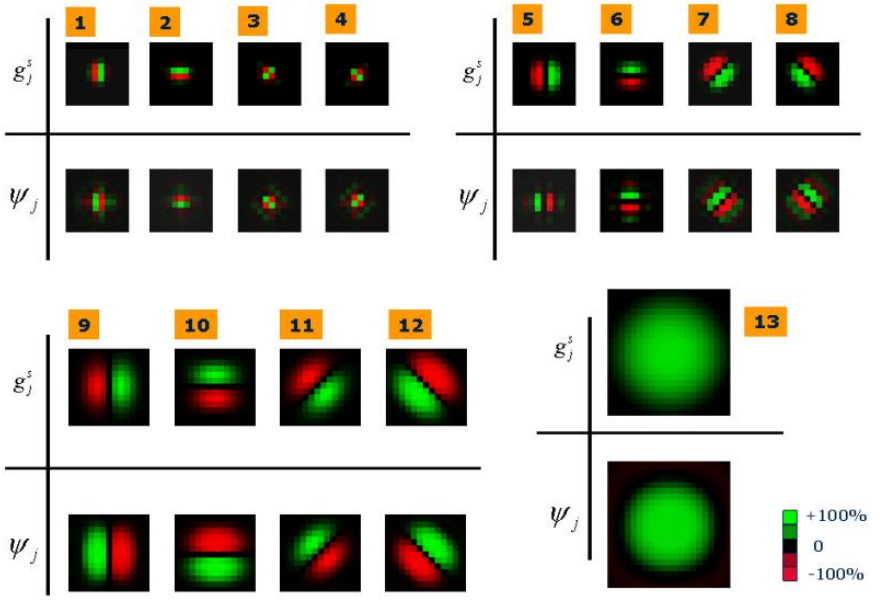


Fig. 9.15 Used wavelets and numerically obtained inverse wavelets as well. Together with the averaging, radial-symmetric wavelet no.13, a total of four resolution layers emerge.

Neural Implementation

The architecture of the net is directly derived from (9.7). We remember that the Gabor wavelets m_j are composed of a positive portion m_j^+ as well as of a negative portion m_j^- (cf. Figure 5.11). The same holds for the inverse Gabor wavelets ψ_j. Then, (9.7) reads:

$$I = \sum_{j=1}^{n} \left((m_j^+ - m_j^-) * (\psi_j^+ - \psi_j^-) \right) = \sum_{j=1}^{n} \Big(m_j^+ * \psi_j^+ + m_j^- * \psi_j^- - \left(m_j^- * \psi_j^+ + m_j^+ * \psi_j^- \right) \Big) =: \left(I^+ - I^- \right) . \tag{9.9}$$

Figure 9.17 shows the neural architecture derived from (9.9).

The left part of the architecture represents layers m_j which are broken down into a positive portion as well as a negative portion. Layers m_j reflect the features g_j contained in image I which are extracted on a 1-pixel grid. Subsequently, the convolution is executed by spreading pairs of input neurons of pulse difference nets on the receptive field (red square) according to the profile of the inverse wavelet. In doing so, both image portions I^+ and I^- emerge and are connected to a final pulse difference circuit to generate image I.

Experimental Results

Finally, we verify the functionality of the proposed architecture using 3 examples (Figure 9.16). The images contain 32x32 pixels, and the set of wavelets as well as inverse wavelets depicted in Figure 9.15 is applied.

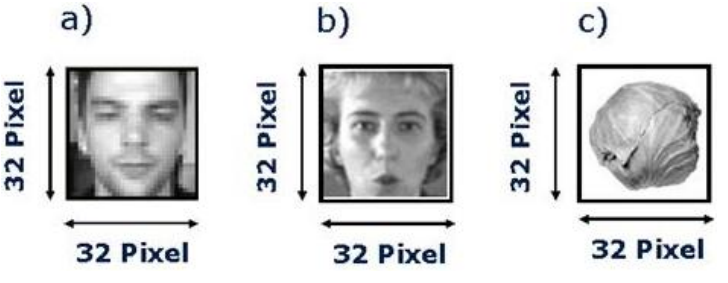


Fig. 9.16 Test images of 32x32 pixels

Figure 9.18 shows the pulse responses of feature detectors situated in layers m_1 by m_{13} for Figure 9.16a.

The simulation time was set to 2.5s for all experiments. For the purposes of the illustration, the pulse rate of neuron N_4 located in the last difference stage in Figure 9.17 is converted to an equivalent grey value according to:

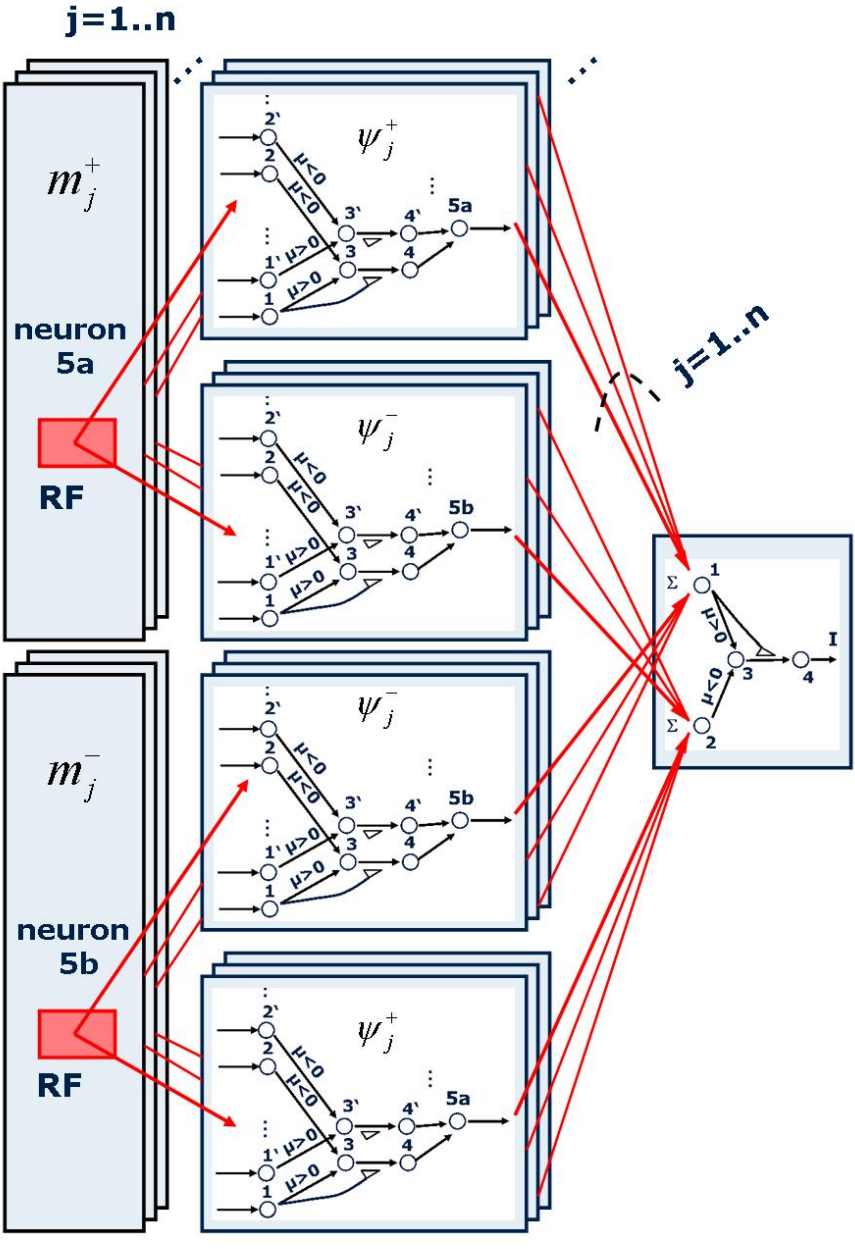


Fig. 9.17 Neural architecture for the implementation of (9.9)

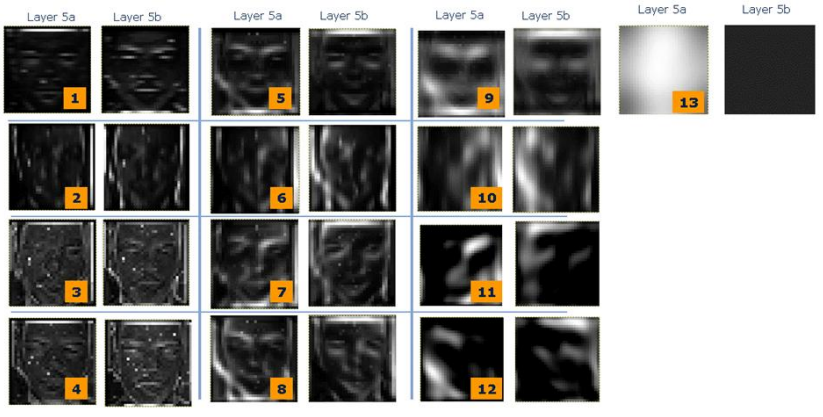


Fig. 9.18 Extracted features for test image Figure 9.16a, layers m_1 to m_{13}.

$$I_R(x,y) = \frac{\Theta}{w_0} \cdot \frac{\pi_4(x,y)}{1 - t_d \cdot \pi_4(x,y)} \ . \tag{9.10}$$

Additionally, the grey value intensities were scaled with respect to the overall maximum value of the layer.

The synthesised image is shown in Figure 9.19 for the progressive addition of partial images. The last column displays the image when synthesised from the complete set of 13 wavelets. The upper row shows experimental results for the neural net given in Figure 9.17 while the lower row shows the mathematical calculations.

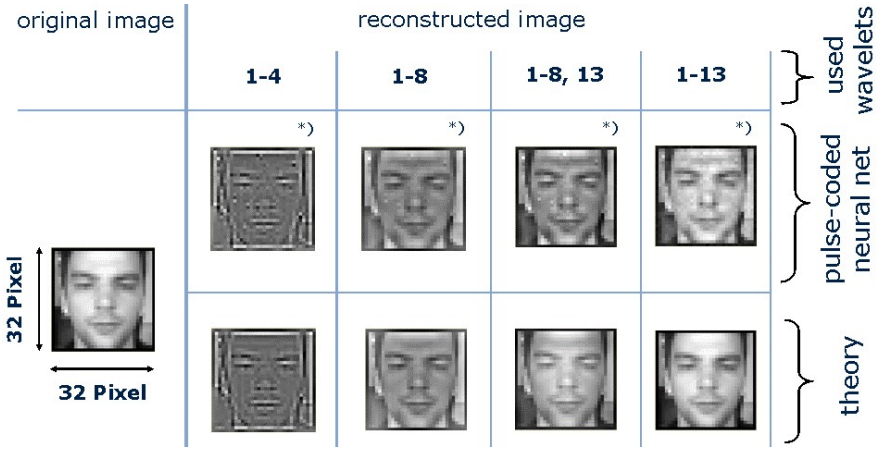


Fig. 9.19 Synthesis results for test image Figure 9.16a: Comparison of theory (lower row) and experiment (upper row).

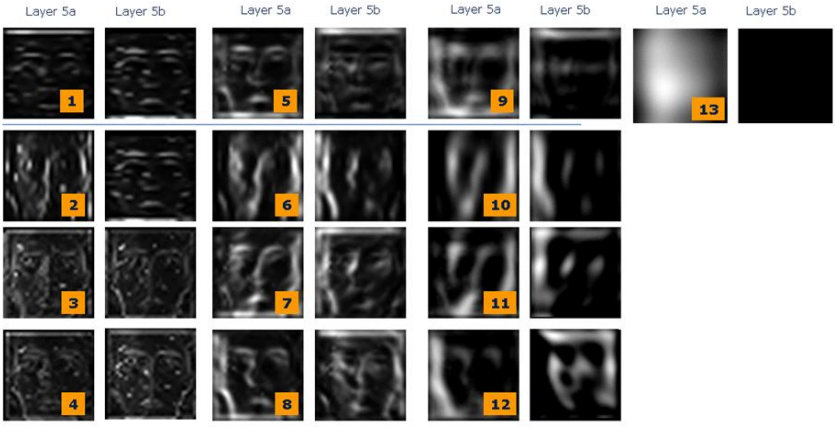


Fig. 9.20 Extracted features for test image Figure 9.16b, layers from m_1 to m_{13}.

Likewise Figure 9.20 and Figure 9.22 illustrate the output activities on the wavelet layer for test images Figure 9.16b and Figure 9.16c. Figure 9.21 and Figure 9.23 depict the respective reconstructed test images. Apparently, there is a good match between theory and experiment.

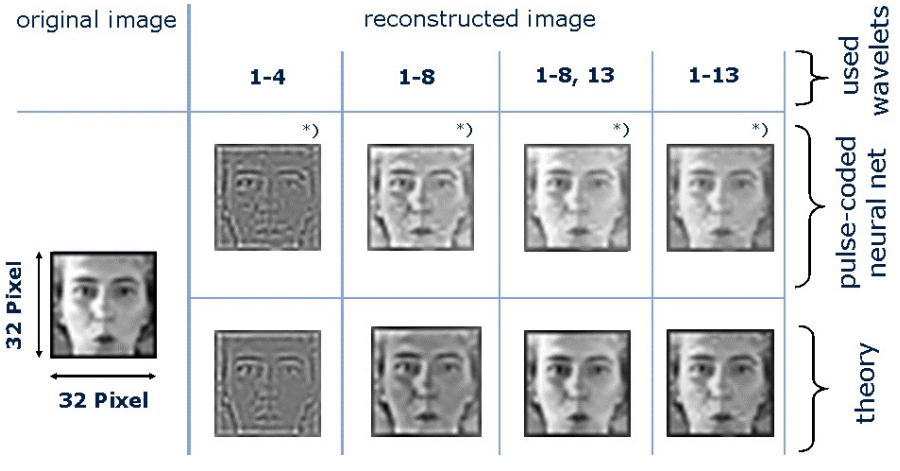


Fig. 9.21 Results from image synthesis of test image Figure 9.16b: Comparison of theory (lower row) and experiment (upper row).

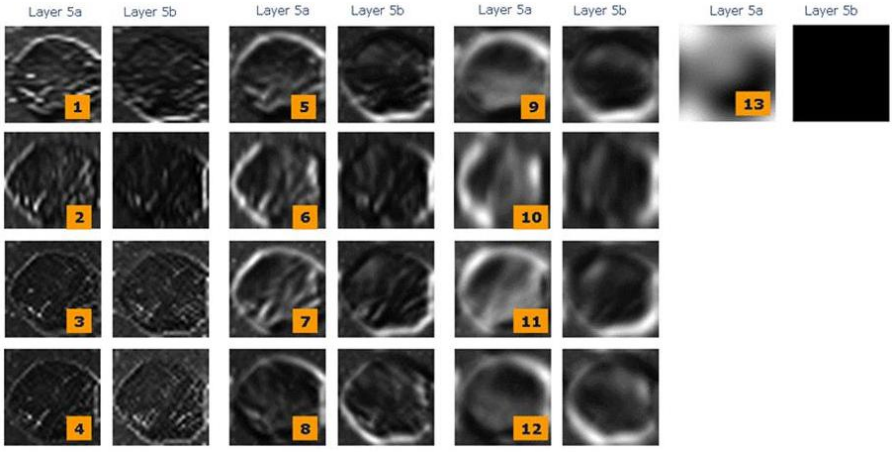


Fig. 9.22 Extracted features for test image Figure 9.16c, layers from m_1 to m_{13}.

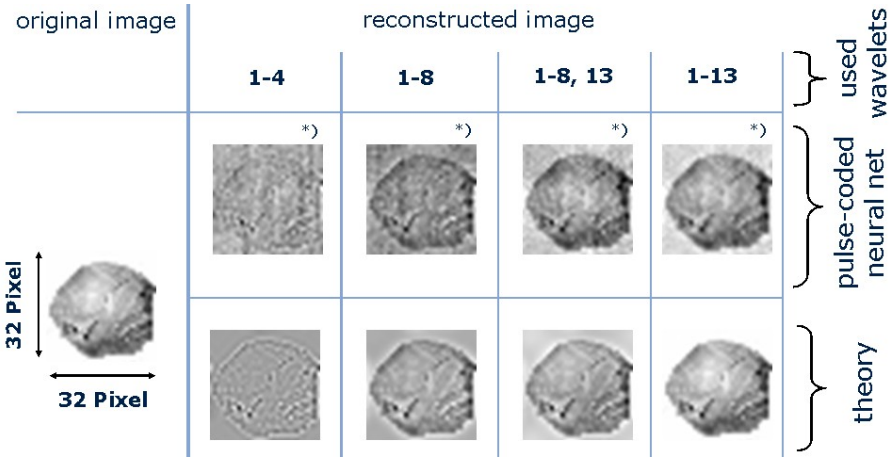


Fig. 9.23 Results from image synthesis of test image Figure 9.16c: Comparison of theory (lower row) and experiment (upper row).

Reconstruction of the $32 \cdot 32$ test images was based on 3 resolution layers and 4 wavelets per resolution layer. Simulation on a PC (2 GHz, 2 GB main memory) required 14 days of computing time per image. The above results allow us to expect satisfactory reconstruction quality when the feature cascade proposed in the last section (comprising 5 resolution layers and 8 wavelets per resolution layer for images with at least $97 \cdot 67$ pixels) is used.

Irrespective of the applied feature cascade - the „inner screen“ requires twice as many neurons and synapses as the feature cascade. Therefore, overall effort remains below 1 percent (see previous section).

9.3 Summary

The head detector architecture presented in chapter 8 was based on a horizontally oriented single feature detector. For a distance-invariant recogniser the number of resolution layers must be increased to 5 and the number of wavelets per layer to 8. As a result, 0.14 percent of the overall number of neurons and 3.45ppm of the total number of synapses in the brain are required to implement the feature cascade.

A direct proof of distance-invariant recognition of heads via simulation was not possible due to limited computing power, since the input images should be at least 97 x 67 pixels in size. On the other hand, the distance-invariant feature cascade has only 4 more layers comprised of smaller wavelets compared to the feature layer used for head detection at the reference distance (cf. chapter 8). Most likely, the desired function is given.

Since the feature detectors occupy an area of at least 2×4 pixels, requiring many feature detectors to participate in recognising a contour (when grasping an apple, for instance,), and since we are able to recall images from memory before our mind's eye, we need a procedure set up at the feature detector level that provides pixel-accurate statements of object positions as well as components and contours. The simplest and cheapest solution seems to involve the construction of a two-dimensional „inner screen“ that displays an image as similar to the retinal image as necessary.

In much the same way that the feature image is generated by the convolution of the retinal image with wavelets, the retinal image is reproduced in high quality by the convolution of the feature detector layers with the inverse wavelets. This means that - except for changed profiles for the distribution of sampling points - the feature cascade can be reused to reconstruct the retinal image. Evidence of successful image reconstruction was provided for images with $32 \cdot 32$ pixels to keep simulation time for the overall architecture including a 3-layer feature cascade within reasonable limits.

Considering that first, the feature cascade is based mainly on pulse difference circuits followed by pulse averaging circuits, and second, that extremely different functions such as feature extraction and image reconstruction can be implemented using profiles that differ only in their „sampling points“, the real value of this basic neural circuit cannot be overestimated, since information received through the senses must first pass through the feature filtering stages and subsequently through the reconstruction stage.

When recalling images from memory and projecting them onto our mind's eye, the question arises as to the appropriate number of zones per object. The more precisely we remember or see an object, the more zones must be

included. On one hand, zones are inexpensive, since only a few neurons and synapses per zone and object component must be implemented. On the other hand, an increase in the number of zones per object requires further extensions of our vision architecture. Let us assume that 100 zones will cover a head. If the object unfolds its dynamics in motion or if the object turns a little bit, the zones' distances from each other will change. In our architecture, a zone by definition covers the largest area of the appearance of an object's component (e.g., the zone representing a mouth within a head). Here, it will be comprised of many smaller zones. However, a hierarchical approach may come into play. We expect to present such a hierarchical solution in the next edition.

In addition, the next most urgent tasks to solve are:

- the binding of variable zones in order to recognise objects such as chairs,
- the construction of an automatic process creating zones.

When these tasks are mastered, a wide variety of different objects can be robustly recognised. Of course, some compositions of the elementary components are still lacking, such as *associative memory*, for instance. But a new field of science, such as the construction of artificial brains, takes more than a decade to mature.

10

Look-out

Christoph von der Malsburg

This book aims at a better understanding of the animal or human brain's function. We are using a constructive methodology, by modelling paradigmatic visual functions with the help of concrete neural models. At the same time we would like to lay the foundation for a new form of information technology according to the biological model. As the functions we have described so far are only a beginning, we here discuss how this approach may be systematically continued. Two issues need particular attention: the interpretation of physical brain states as data format of mental states and the mechanisms of self-organisation and learning. To illustrate the proposed concepts we discuss the problems of invariant object recognition (beyond those treated in chapters 7-9) and of visual scene representation.

10.1 Data Format of the Brain

As to the data format of *memory* we agree with the general assumption that it resides in the neural connections in the brain. These are the protocol and sediment of past brain states, evolve through learning and self-organisation and act as boundary conditions shaping active brain states.

Concerning the data format of the rapidly changing *brain state* there exists a dominant dogma, which we criticize, however, as grossly incomplete [21]. This dogma has it that the brain state is *completely described* by the firing level of individual neurons. If neural activity was all, however, the world represented in our brain would be restricted to objects and situations taken from a limited catalogue of elementary symbols. What is needed instead is a data format that can represent the whole richness of our mind's universe, starting with the constantly changing situation in front of our eyes. This calls for structured aggregates of symbols that in turn are combinations of simpler elements. In one word, the data format has to generate variety by combinatorics, and combinatorics is possible only by the structured attachment of elements to each other.

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 173–182, 2010.
© Springer-Verlag Berlin Heidelberg 2010

To generate, for instance, the mental image of a cellular phone, it is necessary to instantiate components such as surface, color, edge, display, key, alphanumeric symbol etc., and these components have to be endowed with specific values of form, position etc. and with representations of spatial arrangement. In addition, it is necessary that the whole object in turn can be treated as a unit in a larger context. In this way it is possible to implement the entirety of mental phenomena as rapidly changing, hierarchically structured complex symbols, in which a great variety of elements is not only active (as implied by the classic dogma) but also bound together in a structured fashion.

Such hierarchically structured entities may be concretely formulated as dynamical graphs, a formulation that conforms well to the structure of neural and synaptic tissue. Graph nodes correspond to the neurons or groups of neurons of the classical data format. Over and above that, however, a dynamical graph has edges, or *links*, that are activated and deactivated on the same temporal scale as the nodes. Thus, a node *key* in our mobile phone example can be connected through active links with a node for a *position x* on the telephone's surface and with nodes for *quadratic form* and *symbol 1*, while these in their turn may acquire further substance through links to finer structural elements.

On the finest level of complexity, nodes and links are implemented in the brain directly as neurons and synapses. On levels of higher complexity, nodes are themselves already structured networks, and graph edges are realised as structured mappings between these networks [46]. Experimental knowledge on the one hand and ideas of assumed interrelationships on the other hand are to be substantiated differently on different levels of abstraction.

According to this picture, both memory and active brain states have the form of dynamic structured networks, of active graphs, which are a very expressive data format. The slowly changing, „permanent" synaptic weights represent memory and constitute a structural framework within which the active brain state is organised by selective activation of nodes and links under the influence of external signals and synaptic interactions. We now will have to turn to the consideration of these processes of organisation of memory and brain state.

10.2 Self-Organisation

In chapters 5-9 we could have applied the term self-organisation to the phenomenon of synchronisation: local firing events generate coordination of neural firing on the scale of the whole network. When we here apply the term to the self-generation of dynamical graphs composed of neurons and synapses, we put emphasis on purposeful information processing and the pursuit of goals, which are the basis of the brain's ability to autonomously react to external stimuli or to act for a purpose, in other words, to organise itself. This is in contrast to the extraneous organisation of the computer by programs.

Admittedly, the analysis of self-organisation of neural nets on a concrete level leaves many questions open, and we have to restrict ourselves to approaching dynamic graphs on an abstract level for higher vision functions than considered in this book. Of prime interest are the mechanisms of self-organisation of graphs on the fast time scale of brain function. The visual recognition of objects, to take a typical process, takes a mere 100 msec [47]. As demonstrated in chapters 7 and 8, the definite pulse response of an already structured graph for image recognition at good illumination needs ca. 200 msec. For the generation of a new dynamic graph (in the language of modelling) or of a synaptic network (in the language of implementation) much more time is needed. If a novel object is to be perceived, for instance, the dynamic graph is partially pre-defined as it makes use of the feature cascade, but the generation of the essential features is still required. This part of the graph must self-organise. We assume this self-organisation to be based on pre-formed functions or networks. (For mechanisms for the rapid re-activation of previously formed graph structures see the section on invariant object recognition below.)

On the most detailed level, all of this has to happen in terms of the known, elementary processes of network self-organisation: links produce signal correlations between the nodes, signal correlations in their turn change the links through mechanisms of synaptic plasticity. In this way, initially unstructured or poorly structured graphs are modified until a stable configuration is reached, and stabilised in terms of a general objective function such as entropy or correlation. The known mechanisms of synaptic plasticity strengthen successful links: excitatory synapses are rewarded by the success of incoming spikes to excite the target cell within a short time (spike-timing-dependent plasticity [48], as formulated in Eq. 3.6).

It is an interesting problem how to characterise the universe of structured graphs that is defined by such plasticity rules and in what way this pattern universe can serve as a general basis for cognitive processes. According to a hypothesis formulated earlier [21], at least a large class of organised networks in the brain form topological graphs. (A topological graph can be embedded in a low-dimensional space such that all connections are of short range.) For useful applications of two-dimensional topological graphs see the section on visual invariance.

Among the essential properties of our environment are localisation of interactions in space and time, embedding of specific structures in fine-to-coarse hierarchies and a considerable amount of self-organisation in the world that surrounds us. It therefore is not surprising that self-organised graphs with a preference for topological structure reflect these properties of the environment. That such pre-stabilised harmony is actually necessary is a consequence of the no-free-lunch theorems [49], according to which any learning or optimisation mechanism is bound to fail in an uncongenial environment.

10.3 Learning

As we have argued in chapter 1, genetic information doesn't suffice to structure the brain freely. But neither does the information acquired through learning. According to psychologists we pick up less than one hundred bits of information with a single sensory impression. If each sensory impression takes a second, then during a lifetime of less than 2×10^9 seconds we can pick up only $2x10^{11}$ bits of information (even this a generous estimate), that is, much less than the $7x10^{15}$ bits of information contained in the cerebrum's wiring pattern. Also this shows that the brain must restrict itself to generic, self-organised structures.

Up until today, the mechanism of autonomous learning from the natural environment is an unsolved riddle. The essential points still to be modelled are that, first, significant patterns must be selected from the sensory data - that is, patterns of future value - and that, second, these patterns must be deposited in a structured fashion in an appropriate place in the memory. To solve these two problems, a learning event must be initiated by the recognition of segments of the sensory field against schemas pre-defined by evolution or previous learning. Segments are thus thematically identified and can be directed towards a small segment of the brain to be modified. (The latter idea is also the basis of the mixture-of-experts model [50].) The principles of statistical learning can thus be applied. As discussed in chapter 1, these are effective only if relatively small and thematically homogeneous patterns are fed into a system, and if this system is small. Grossberg's plasticity-stability-dilemma [51] - the conflict between learning, which needs plasticity, and preservation of old memory traces - can in this way be solved, as only recognised neural patterns are addressed, the rest of the memory being left undisturbed.

The selection of parts of the sensory and the memory domains of the brain during the learning process is achieved by focal attention. Attention does not imply suppression of activity in the rest of the brain [52]. Recent experimental results [53] suggest, however, that the focus of attention is defined by oscillations in the gamma range (40 - 80 Hz). One may surmise that the focus of attention selects from all currently active neurons a subset that stands out by particularly strong momentary coupling that causes the neurons in this subset to engage in coherent oscillations. The spot detector described in chapter 5 may perhaps be seen as a special case in this process. The concentration of neural pulses into narrow time intervals caused by these oscillations may play a decisive role in the induction of synaptic plasticity. After a moment, neurons and links within the focus of attention fatigue, so that a new set of cells and links can pick up the ball.

The discussions in this chapter up to here were of abstract nature and need to be illustrated by concrete applications. We will undertake this in the next section.

10.4 Invariant Object Recognition

A very large class of patterns to be picked up during learning are of a visual nature and relate to concrete objects or scenes from our immediate environment. These patterns strike our brain in the form of retinal light distributions. One and the same object may generate an infinite variety of possible images, according to differences in position, size, orientation, illumination and so forth. This variety is to be reduced as far as possible by transformation of the images into an invariant normal form. In chapters 7-9 we have described concrete architectures for the recognition of human heads while varying illumination, distance, form and, to a limited extent, rotation. For more substantial variations - position, rotation in the image plane - this transformation is possible in an object-independent way with the help of rapidly switching fiber projections (*dynamic links*, [21, 54, 55] or *shifter circuits*, [56]). On this level a simple system could, in schematic form, be structured as in Figure 10.1. A segment of an *image domain* (primary visual cortex) is projected by a system of dynamic links on to an invariant window.

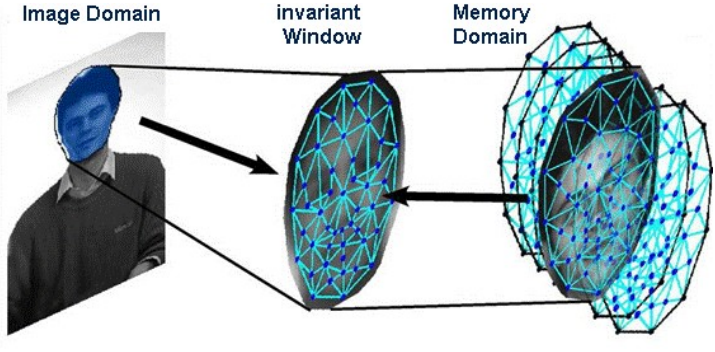


Fig. 10.1 Principle of invariant object recognition: A segment of the image domain (left) is transferred by a temporarily activated fiber projection to an invariant window (middle). This invariant representation of the segment is compared with object models stored in a model domain (right). The segment is recognised by finding a model with high similarity.

The image segment, after being represented invariantly (that is, in a normed position, size and orientation), can now be matched to stored object models. The segment is recognised if a model with high similarity is found. After each eye movement or image change, a new dynamic link projection has to be established. The structure of an image segment (or of the whole image domain) is that of a two-dimensional, locally interlinked array of feature recognition cells (as in the object recogniser described in chapter 8). After an object is successfully recognised, the activated fiber projections between the image

domain and the invariant window, and between the latter and the model in the model domain, have the form of homeomorphic mappings, linking cells of the same feature type while preserving neighborhood relations. If the image of the object in the image domain is of non-standard size or orientation, the connections between feature cells must also be altered [57] (if the features are not themselves invariant). Since such variable fiber projections are able to compensate for local deformation, and since some aspects of the illumination problem can be solved by signal normalization, a large part of the invariance problem is thus solved. An object recognition system that conforms to these principles, which has been realised in concretely neural form (if not in the amount of detail emphasized in this book) and which achieves respectable face recognition rates on large data bases is described in [58]. The model of a face finder presented in chapter 8 covers part of the functionality described here, (restricting itself to the relation between the invariant window and a model domain containing just one model), but in exchange realises this part in full neural concreteness.

It is important to observe that the network structures employed for realising the object recognition system are all of the topological type (see the last-but-one section). For one thing, the representation of images or image segments in the image domain has the form of two-dimensional networks with short-range connections (a fact that rests on a solid experimental basis, cf. also chapters 7-8). This applies also to the representation of image segments in the invariant window and of the objects in the model domain. Once these representations are, at the end of the recognition process, tied together point-by-point through temporarily activated link structures in a neighborhood-preserving fashion, the resulting comprehensive structure is likewise in the form of a topological graph. This structure is an attractor in the sense of the synaptic plasticity rules discussed above.

In principle, the neighborhood-preserving fiber projections necessary for object recognition could be activated time and again by *ab initio* self-organisation, as modelled in [59]. This would, however require several seconds, a hundred times slower than human object recognition [47]. It therefore is to be postulated, as hinted at above, that the ordered fiber projections are formed and stored [60] at an early stage of development (perhaps even before birth), so that during the recognition process it is merely necessary to activate the relevant pre-formed connectivity structure. Object recognition can thus be achieved very rapidly, in under a tenth of a second [58]; cf. also chapter 8).

As mentioned above, pattern recognition is a pre-requisite for learning. Conversely, learning is important for recognition. At birth, the system is endowed with mechanisms and search patterns of great simplicity to boot-strap this interplay. Initially, these mechanisms may work only in simple situations, but in a self-amplifying process they steadily improve by learning. Among the first patterns recognised as significant may be images of objects that move against a static background; these can be identified and singled out by simple temporal filtering. However, over time figure-ground segmentation has to be

gradually improved with the help of plastic modification of horizontal connections in the image domain and through the availability of better and better schematic descriptions of potentially significant patterns. Likewise, feature detectors and their connections are improved through learning, and eventually a growing and ever more complete repertoire of models and model pieces is accumulated by learning.

In this section and in chapters 7-8 we assumed that in the model domain each object is represented as a separate network covering the object in its entirety. Such monolithic representation is well suited for certain classes of objects, among them faces; the majority of types of objects, however, has to be flexibly composed as a collage of model parts [61]. We therefore should now have a closer look at the structure of the memory domain.

10.5 Structured Memory Domains

Our visual system scans the scene that surrounds us with focused attention and recognises familiar patterns. As stored patterns have, after many years of experience, accumulated connections to other modalities and have thus acquired meaning (name, form, texture, material, functional role, appropriate manipulative motion patterns etc.), the scene is analysed and interpreted by activating these stored networks. The reality that we perceive and identify with our environment is, of course, happening inside our head. Infinite detail seems to be available to our intuition and all this detail is filled with meaning by us. This visual reality is in its substance to a very large part a collage of structural pieces called up from our visual memory. To make this possible, our memory constructs a model kit in the course of our visual experiences, an immense mass of pattern representations, which can, in the most flexible way, be combined to render the structure of our environment.

During the inspection of, e.g., a chair, the system must have the flexibility to respond to a large variety of forms, materials and viewing angles (Figure 10.2). It does so, as shown convincingly in psychophysical experiments [62], by deconstructing the image into familiar structural parts (bricks, plates, corners, surface patches, etc.) and spatial arrangements. The system recognises these parts and connects them under deconstruction with the help of dynamic links with the current invariant representation of the chair and with each other. Deformation is required as the parts vary in their dimension and foreshorten and change their form according to the point of view. Each activation of parts and their linking to the invariant segment representation is a process of recognition, exactly as described in the last section. The structural parts are two-dimensional topological networks composed of feature-bearing cells, whose distribution gives the part a unique aspect.

Among the features are form characteristics, such as local edges, brightness gradients, etc. Each structural part can deck itself out with further representations - for instance, color distributions and textures - again with the help

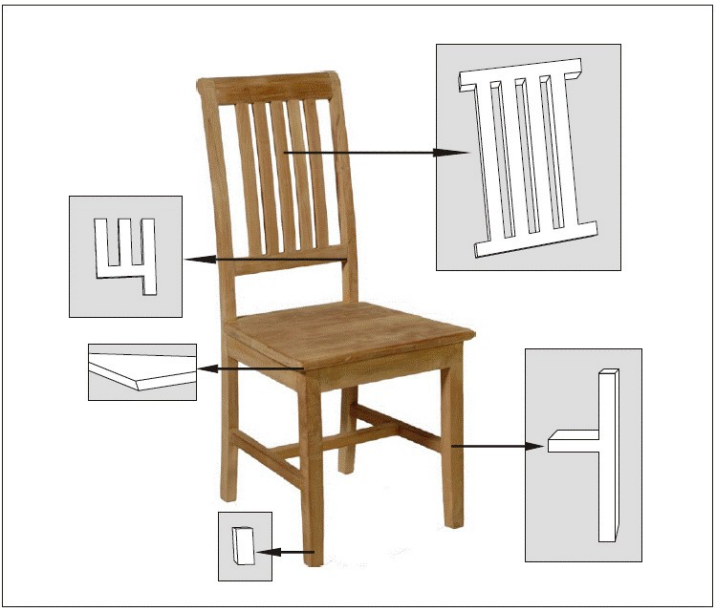


Fig. 10.2 Object as collage of structural parts, on the example of a chair. (A full covering would involve many more parts, with a different scale and level of detail, multiply overlapping each point.)

of dynamic links. Each known three-dimensional part of the environment is represented in the memory by several networks in order to give different viewing angles. These representations are connected with each other through a network structure on a higher level, so that when the chair rotates in front of our eyes, the representation of each part is exchanged in a coordinated fashion to reflect the varying viewing angle.

In this way, the representation of the invariant image of an object - such as a concrete chair seen under a certain perspective - is a complex, hierarchically structured network, composed of sub-networks, each of which is acquired by experience, stabilised through network self-organisation, and connected with each other through homeomorphic mappings. Among these mappings those have a special status that connect the memory domain through the invariant window with the image domain and thereby with the object out there. The role of the image for the generation of the representation in the memory domain is, however, not to be seen so much as the generation of a copy but rather as a process of guided Gestalt formation - a little like a conductor light-handedly leading the orchestra or the genes modulating the process of morphogenesis in the embryo.

The representation of a chair in its concrete visible form is one thing, its functional involvement in the planning of some action is another. It is the

purpose of a chair to provide me with a surface of convenient height, form and size within reach so I can sit on it (properties that may be used for defining a chair; Gibson speaks of *affordances*, opportunities to perform an action or a manipulation). It is therefore important that the structural parts of an object representation connect themselves in an appropriate way to the networks responsible for the planning of action in my brain.

The recognition of a chair helps build up the representation of a scene. Conversely, however, the representation of the scene in its entirety helps to recognise the chair, as chairs are likely to be found in certain regions of certain scenes. The basis for orienting ourselves and for planning our movements is the representation of the entire scene surrounding us. We may imagine this representation to be based on a *canvas*, of a two-dimensional network of point tokens, each of which stands for a spatial direction as seen from my point of view. The canvas would, according to this idea, be populated in the course of the recognition process with symbols that are affixed at their respective points or regions. These symbols may refer to objects or to types of surfaces, such as *floor*, *wall*, *landscape* or *sky*, but may especially also carry information regarding the distance and orientation of surfaces to support the precise performance of movements [63]. Symbols, that is, the cells and networks that represent the recognised material, are attached to the canvas with the help of activated links. According to neurological and neurophysiological evidence, it is likely that the canvas is in the parietal cortex (referred to in this context as the *where* or *how pathway* [64]), whereas the recognised objects are in the temporal cortex (*what pathway*). For our subjective impression, the surrounding scene is simultaneously available to our inner eye, each single blade in the meadow before us, for instance. This impression I can confirm any time experimentally: I flash my attention to any location whatsoever, and behold the detail is there. The impression of the simultaneous presence of all detail is, of course, an illusion, since as soon as I change my attention and gaze, everything sinks back into nothingness or at best into the form of an abstract shadow of the thing I saw, affixed by activated links to the scene canvas.

10.6 Summary

To understand the brain's structure and function and to prepare for its reconstruction, several fundamental issues are to be addressed. Among these are the data format of memory and of the brain state as well as the mechanisms that organise these data structures. What is not required is universality - the passive readiness of the computer to implement any human-designed structure - but the ability to acquire structure with the help of self-organisation and learning. A highly versatile data format is the graph, whose nodes act as elementary symbols and whose links are a basis for combining these elementary symbols in flexible ways into complex symbol structures. Graphs are generated by network self-organisation, whose mechanisms form a universe of structured

nets. Part of this universe are topological networks, which are distinguished by being dominated by short-range connections if embedded appropriately in a low-dimensional space. These topological graphs are ideally suited for vision applications, in that images can be represented as locally linked two-dimensional nets, and in that the recognition process and the construction of composite object models is possible with the help of homeomorphic point-to-point mappings. The recognition and representation of objects and scenes is thus possible with the help of a model-kit of structured networks that dynamically interlink in a process analogous to crystal formation.

The application of these ideas is not limited to the vision process. Other brain processes may proceed according to similar principles and also a great variety of information-technological applications might possibly be formulated in a similar way.

11

Preliminary Considerations on the Microelectronic Implementation

Ulrich Ramacher, Jens-Uwe Schlüßler and Jörg Schreiter

11.1 Equivalent Representations

The simulation and calculation of the activity diagrams in Figures 8.2 a-n required about 10 days and nights. A statistical testing or significant augmentation of the architecture by means of simulation on a PC is therefore not a passable route in the long run. An essential advancement is offered by these two paths:

1. the creation of equivalent representations for the constituent steps of the recognition chain,
2. the microelectronic implementation of synapses, neurons, and whole nets.

The first path has not yet been dealt with systematically, so we are confining ourselves to some hints. The first component of an object detector is the feature detector, which was given the characteristics of a Gabor wavelet type. After having proven the satisfactory agreement between mathematical and neural Gabor wavelets in chapter 6, we can replace the actual transfer behaviour of the implemented detector, including its upstream pixel-pulse converter, with the pixel-controlled transfer behaviour of the mathematical detector, and subsequently convert the analogue values into a pulse rate. However, this would not result in a significant reduction of the simulation time. A better solution is obtained by substituting the neuron complex N_1-N_4 with a difference counter. This expedites the simulation dramatically.

The transfer behaviour of the subsequent feature recogniser could be copied by searching for the largest connected set of neurons having equal pulse rates. Here any segmentation technique can be used. However, these require quite intensive calculations, and an analogue implementation of our synchronisation detector always seems to be more advantageous.

On the other hand, the binding of several feature types to object components or of object components to objects is performed best with digital logic.

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 183–194, 2010.
© Springer-Verlag Berlin Heidelberg 2010

An equivalent representation should always reduce the simulation time by at least 1 order of magnitude, compared to the reference implementation of simulating neural signal equations, in order to provide a guide for advanced experiments testing the next steps of a vision architecture.

Computer-vision based algorithms, thus, can constitute an element of constructing artificial brains provided they adhere to the required equivalence. Accordingly, analogue as well as digital implementation styles shall be applied in future for artificial brains.

For this approach it is crucially important that a synaptic-neural architecture of a vision step exists before the construction of an equivalent function and representation, resp. Otherwise computer vision algorithms would be introduced which lack a functional realisation by neurons and synapses, ie. arbitrariness will infiltrate the construction of artificial brains. Note that the last 50 years of computer vision research teach us that vision can not be understood and human performance cannot be reached without the neural approach to vision.

11.2 Microelectronic Implementations

The simulation problem has been our permanent companion in chapters 2-9. It has been emphasized in the last chapter that we only begin to understand the brain, and many more features have to be added to the visual ones - in particular non-visual ones - so that we can arrive at the abstractions necessary for object recognition. Assuming that these additional feature modules have been constructed, their final test has to occur in relation to the whole vision net. The connectivity and total size of a complete neural net, thus, may increase faster than the performance of computers and super-computers.

Even the creation of equivalent representations will not significantly lessen this problem, so that other solutions must be envisaged. For the moment, the sole one seems to be the electronic implementation of dynamic synapses and integrate-and-fire neurons on the basis of microelectronic or nanoelectronic circuits. Here we see two challenges: Firstly, the implementation of the signal equations for neurons and synapses in the sense of equivalent representations, and secondly, the design of the system architecture.

In the first case we have to focus on the adequate realisation of the courses of the functions on a value or time basis by means of analogue circuitry. The achievable realisation will never be ideal, and the closer it approaches the ideal, the higher the costs will be. Small deductions have to be made for the implementation of integrate-and-fire-neurons, whereas dynamic synapses require considerable work when the differential equation must be implemented with an accuracy comparable to 8 bit or more. The key to an optimal implementation is found in reflecting on the robust behaviour of the dynamic synapses: with a pulse being present, the value of the synapse is to increase or decrease, depending on whether the membrane potential lies above or below

half of the threshold value. Similarly for a dendritic synapse. We had realised this behaviour with a differential equation, but we could have chosen some other rule-based formulation, as well. In this respect, our differential equation should not be valued too high. It is a comfortable formulation of the described synaptic behaviour that allows for efficient visual processing and a simple connection of theory and experiment. First, a biophysical analysis will show how good our differential equation matches the real biological processes or to what extent it abstracts these processes[1]. Anyway, the tolerance as to the way of formulating the synaptic behaviour should be taken advantage of by the microelectronic implementation.

In the second case, it depends on whether the natural pulse rate range of a few ten to hundred Hertz can be transformed into the kHz to MHz range so that the events occurring in the recognition chain happen to run some ten thousand times faster than they did originally. This way, the electronic neural net can emulate thousands of neural nets, which will result in lower system costs. Let us remember the zone architecture of the feature cascade, for instance. There, each feature must be repeatedly implemented within a 2-dimensional grid covering the retina. The grid can be scanned with a clock frequency in the MHz range and the data stored in a memory from which they are input into an electronic realisation of neurons and synapses that implements the Gabor wavelet filtering (chapter 6). A further increase to over 100 MHz may turn out to be counterproductive as the synchronicity of events in the net may no longer be guaranteed.

Besides the design criteria of area and power consumption, which have already been mentioned, flexibility and reuse must be taken care of. Automatically FPGA-like structures come to mind: instead of logic gates, we would need analogue circuitry for neurons and synapses, some multiplexer and sample-and-hold devices, and finally some digital overhead for control of connections to inputs or other neurons. However, when defining basic modules, one is well advised to look at the recognition chain first.

In the following chapters, we firstly present basic circuits for neurons, synapses, photosensors. Then we assemble larger blocks for feature detection and recognition, etc. For this we will use 130, 90 or $65nm$ CMOS technology. So the components of the recognition chain as described in chapters 5-9 become available.

Because of its simplicity, the first basic module to be presented is the feature recognition unit discussed in chapter 7. It is completely analogue, and despite the roughly approximated synaptic behaviour the synchronisation of the neurons happens in full agreement with the simulations, provided sufficient neurons receive the same pulse rate, in which case a pulse wave speeds over the chip. Incidentally, we prefer to use digital circuits to realise the binding of diverse features to an object component, or of object components to an object.

[1] See also last page of this chapter.

Secondly, there follows the description of a chip with the basic module for feature detection as in chapter 6. It turns out that digital realisation is the most advantageous one in this case. Agreement with simulation is demonstrated.

In chapter 16 a 3-dimensional stacking technology for silicon chips is characterised and a prototype with 6 silicon layers built to test the 3D wiring net. The last chapter presents the architecture of a „vision cube“ which supports a feature cascade for the real-time recognition of 1000 objects on the basis of 8 orientations and 5 resolution levels. For this 59 chips, each of about $60mm^2$, are stacked to a height of $1mm$.

The remaining pages of this chapter are devoted to the description of various behavioural models for neurons and synapses, starting out with those close to biological processes and finishing with the more abstract ones used in this book. With this parade of neuron and synapse models we remind the reader of the need for a workable abstraction.

11.3 Models of Neurons and Synapses

Biological Neurons

The majority of neurons possess a three-membered outer structure: out of the soma sprout one or more strongly branched extensions, the so-called dendrite, and a longer less strongly branched extension, the so-called axon. Electrophysiological measurements of isolated and *in vivo* neurons showed that electrical pulses propagate alongside the axon, which feature an amplitude of about $100mV$ and a duration of about $2ms$. Such pulses are designated as action potentials. The onset of the action potentials is explained by electrochemical processes within the cell and its circumfluent intercellular fluid [65].

Depending on the concentration difference and the electrical field, the ions (Na^+, K^+, Ca^{2+}, Mg^{2+}, $Cl^-$and amino acid rest ions A^-) move through the ion channels of the cell. Particular protein molecules contained in the cell membrane, and which function as ion pumps, take care of the perpetuation of the concentration gradient.

An electrical model of the origin of action potentials is presented briefly in the Appendix 19.2. Accordingly, when inserting a pulse-shaped current - for example, $I(t) = I_0(H(t) - H(t - t_0))$, with amplitude I_0, duration t_0 and H the step function - the membrane voltage reaches up to $35mV$ for just a moment. As for the biological example, a threshold behaviour results: with fixed duration of the input, pulse action potentials will be provoked provided I_0 is stronger than the threshold I_{th}.

Biological Synapses

Synapses create connections between neurons, the pulse (or spike) propagating from the presynaptic cell to the postsynaptic cell. If an action potential

reaches a synapse via the axon, the presynaptic cell releases a substance called *neurotransmitter* into the synaptic cleft, which is the space between the pre- and postsynaptic cells. The substance diffuses towards the postsynaptic side, temporarily bonding at receptor molecules. There are two types of receptors: ionotrop receptors directly influence the penetrability of the respective ion channels, whereas metabotrop receptors generate a cascade of chemical reactions which finally determine the permeability of certain ion channels. The remaining neurotransmitter is partly re-absorbed by the presynaptic cell and partly decomposed by respective enzymes. For more details see [65] and [66].

Postsynaptic Reaction

These processes result in a specific time response for the penetrability of a respective ion channel in the range of the synapse.

Depending on the difference between the membrane voltage and the reversely-directed voltage of the ion current, a so-called *p*ost*s*ynaptic *c*urrent (PSC) traverses the cell membrane. Action potentials arriving at excitatory synapses may cause an action potential for the postsynaptic neuron, whereas in the case of inhibitory synapses existing, postsynaptic action potentials could be suppressed. Accordingly, the current passing through the synapse is called excitatory or inhibitory postsynaptic current (EPSC and IPSC, resp.)

Often the synaptic conductance is chosen for modelling. Many synapses can be modelled by an exponential function,

$$g_S(t) = \widehat{g_S} H(t - t_\mathrm{f})\, \mathrm{e}^{-a(t-t_\mathrm{f})}, \tag{11.1}$$

or by the difference of two exponentials in order to accomodate for the finite velocity in opening a channel,

$$g_S(t) = \widehat{g_S} H(t - t_\mathrm{f}) \big(\mathrm{e}^{-a(t-t_\mathrm{f})} - \mathrm{e}^{-b(t-t_\mathrm{f})}\big) / \big(\mathrm{e}^{\frac{-a\ln(a/b)}{a-b}} - \mathrm{e}^{\frac{-b\ln(a/b)}{a-b}}\big), \tag{11.2}$$

which for $b \to a$ fades to the so-called alpha function [65]

$$g_S(t) = \widehat{g_S} H(t - t_\mathrm{f})\, \mathrm{e}\, a(t - t_\mathrm{f}) \mathrm{e}^{-a(t-t_\mathrm{f})} \ . \tag{11.3}$$

Here H is the step function

$$H(t) = \begin{cases} 0 & t < 0\,\mathrm{s} \\ 1 & t \geq 0\,\mathrm{s} \ , \end{cases} \tag{11.4}$$

and t_f the time when a pulse reaches the synapse and provokes the PSC. The parameters $a \in \mathbb{R}\,\mathrm{s}^{-1}$ and $b \in \mathbb{R}\,\mathrm{s}^{-1}$ serve to fit the time response, $\widehat{g_S}$ designates the amplitude of the synaptic conductance. The latter lies in the range of a few 10 pS for single synapses. Figure 11.1 shows the synaptic conductance according to the equations (11.1), (11.2) and (11.3) for selected values of the parameters.

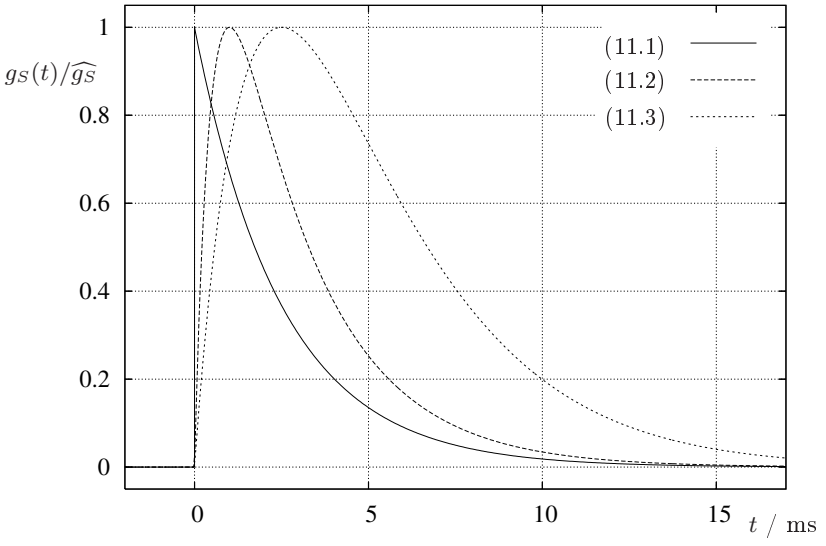


Fig. 11.1 Synaptic conductance $g_S(t)$, scaled to its peak value $\widehat{g_S}$ for $t_f = 0$, $a = 400\,\mathrm{s}^{-1}$, $b = 2\,000\,\mathrm{s}^{-1}$.

A change of the synaptic conductance causes a change of the membrane voltage, which is designated as postsynaptic potential (PSP). Depending on the direction of the current caused by the PSC, we have excitatory or inhibitory postsynaptic potentials (EPSP and IPSP, resp.).

Action potentials at excitatory synapses depolarise the membrane ($U(t) > U_R$ for $t > t_f$). Ions conducted by inhibitory synapses have a Nernst voltage close to the rest potential, often a little below, resulting in an increased value of the conductance of the membrane for a short time. During this time the amplitude of the EPSP created by an adjacent excitatory synapse is diminished, so sliding faster to its rest potential. This form of decreasing the postsynaptic depolarisation is designated as *silent inhibition* and *shunting inhibion*, resp., because of the parallel arrangement of membrane conductance and synaptic conductance.

Some of the models described in literature [66] deal with PSC or PSP in a simplified manner as a synaptic current

$$I_S(t) := U_0 g_S(t) \tag{11.5}$$

and membrane voltage, resp.,

$$U_S(t) := U_0 R_0 g_S(t) \tag{11.6}$$

in the synapse's neighbourhood produced by an independent time-varying source. Here g_S is one of the functions according to (11.1) bis (11.3). The

constants $U_0 \in \mathbb{R}\,\mathrm{V}$ and $R_0 \in \mathbb{R}\,\Omega$ are chosen so to match the amplitude of PSC and PSP, resp.

Synaptic Efficiency

An incoming action potential does not always create a postsynaptic reaction with a synapse. The distribution of the neurotransmitter occurs with a certain probability p [66]. This is taken care of by the value of the parameter $\widehat{g_S}$ in (11.1) - (11.3). An efficiency w for the connection between pre-and postsynaptic cells results from the transmittance probability p, together with the number of synapses spent for a connection to a neuron. On average, the amplitude of the synaptic conductance of the total connection is proportional to the efficiency w. Therefore, w_{ij} acts as the weight of the connection between neuron „i“ and „j“.

Dentritic Tree, Compartmental Models and Point Neuron

Many synapses of a neuron are situated at the branches of the dendrite tree. The tree is often modelled as a passive conduction net characterised by capacitor, serial and shunt resistor values, the latter reflecting type and number of ion channels. Note that the ion channels feature a voltage-dependent transition behaviour between their on and off configuration. Thus, the linear model of the conductive net is an approximation only.

For purpose of numerical analysis the dendrite tree is divided into small segments, sometimes several hundred. Each segment, and also the soma, is modelled by concentrated state variables and, hence, the electrical model by capacities and conductances. The segments are connected by resistors. All these models are called compartmental models.

If the time-variant processes in the dendritic tree are neglected, the synaptic inputs distributed over the tree are considered to be active directly at the soma. The only compartment then is the soma, and the compartment model simplified this way is called the point neuron model.

Spike Response Model (SRM)

Formal models of pulse coupled neurons are guided by the perception of the threshold as described in the Hodgkin-Huxley model (Appendix 19.2). When neglecting the dependence of the triggering process of an action potential on the further temporal course of the membrane voltage after passing the threshold, as well as from the short delay between the action potential passing the threshold and reaching its maximum, the time t_f, at which the membrane voltage U passes the threshold U_th,

$$U(t_\mathrm{f}) = U_\mathrm{th}, \qquad \frac{dU(t_\mathrm{f})}{dt} > 0 \tag{11.7}$$

can be regarded as the exact time of firing. The exact temporal course of the action potential is irrelevant, then, and the output of the neuron is described by the times of firing: $\{t_{\mathrm{f}}^{(n)},\ n \in \mathbb{N}\}$.

A general model for point neurons is described in [65]. The only state variable is the membrane charge Q which is represented by the membrane voltage U because of the linear membrane capacitor. At the firing times, the membrane voltage may fluctuate. The response of the membrane voltage $U(t)$ between two consecutive firing times $t \in (t_{\mathrm{f}}^{(n)}, t_{\mathrm{f}}^{(n+1)})$ is independent of its response in the intervals between other firing times, ie. for $t \in \left(t_{\mathrm{f}}^{(n+i)}, t_{\mathrm{f}}^{(n+i+1)}\right),\ i \in \mathbb{Z},\ i \neq 0,\ i < n$.

Therefore only the last firing time

$$\hat{t}_{\mathrm{f}} := \max\left\{t' \middle| t' \in \{t_{\mathrm{f}}^{(n)},\ n \in \mathbb{Z}\},\ t' < t\right\} \tag{11.8}$$

is to be taken care of at time t. In the SRM, between the firing points the membrane voltage U_i of the i-th neuron is a linear combination of the action potential $U_\eta(t - \hat{t}_{\mathrm{f}})$, including the overshoot when decaying, of the weighted synaptic potentials $U_\epsilon(t - \hat{t}_{\mathrm{f},i}, t - t_{\mathrm{f},j}^{(n)})$, and of the filtered continous input $I_i(t)$:

$$U_i(t) = U_\eta(t - \hat{t}_{\mathrm{f},i}) + \sum_{j \in N_i} w_{ij} \sum_{n \in \mathbb{Z}} U_\epsilon(t - \hat{t}_{\mathrm{f},i}, t - t_{\mathrm{f},j}^{(n)}) + \int_0^\infty \kappa(t - \hat{t}_{\mathrm{f},i}, s) I_i(t - s)\, ds\ . \tag{11.9}$$

Integrate-and-Fire Model (IAF)

A simplified model of a point neuron is shown in Figure 11.2. The membrane is represented by the capacitor C and the fixed conductance by G. The input signal is formed by the current I, which is the sum of the continuous component $I_{i0}(t)$ and the synaptic components $I_{\epsilon,ij}(t - t_{\mathrm{f},j}^{(n)})$:

$$I_i(t) = I_{i0}(t) + \sum_{j \in N_i} \sum_{n \in \mathbb{N}} I_{\epsilon,ij}(t - t_{\mathrm{f},j}^{(n)})\ . \tag{11.10}$$

For this input signal, the membrane acts as a low pass filter with the output signal U_i. In the literature, we also find the name *leaky integrator*.

The firing point t_{f} is given indirectly by

$$U_{\mathrm{th}} = \lim_{t \to t_{\mathrm{f}}^-} U_i(t)\ , \tag{11.11}$$

with U_{th} designating the threshold. Thereby the membrane voltage U_i is reset to $U_0 = \lim_{t \to t_{\mathrm{f}}^+} U_i(t)$.

The reset is typically carried out by a rectangular pulse of duration t_{d} the current of which is limited by the resistor $R_{\min}$. In the idealized IAF model

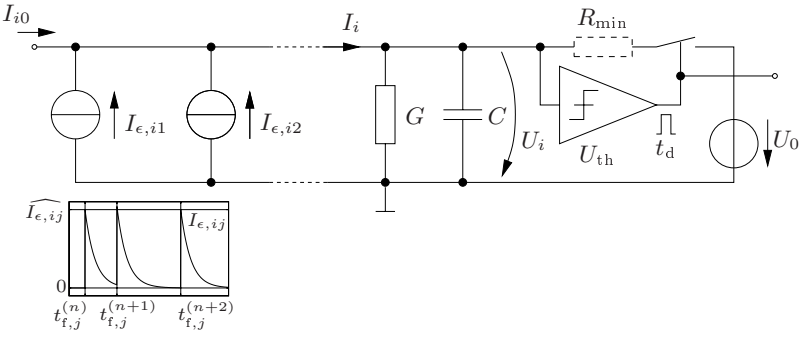


Fig. 11.2 Circuitry of the IAF model

the reset process can be modelled as the limit case for $t_{\text{d}} \to 0$, $R_{\min} \to 0$. Then the output pulse appears as a *Dirac* pulse, and the membrane voltage U_i is reset instantaneously to U_0. Note that the IAF model is absolutely determined by the firing times $\left\{t_{\text{f}}^{(n)},\ n \in \mathbb{N}\right\}$.

The synaptic inputs are realised by independent current sources, the current values $I_{\epsilon,ij}$ of which depend on the time span $t - \hat{t}_{\text{f},j}$ passed since the last firing times $\hat{t}_{\text{f},j}$ of the presynaptic neurons j. Often one chooses:

$$I_{\epsilon,ij}(t) = I_S w_{ij} \mathrm{e}^{-a(t-\hat{t}_{\text{f},j})} H(t - \hat{t}_{\text{f},j}) \ . \tag{11.12}$$

Together with the impulse response

$$g(t) = \frac{1}{C} \mathrm{e}^{-tG/C} \tag{11.13}$$

of the linear membrane model it results a PSP similar to (11.2):

$$U_{\text{PSP}} = g \cdot I_{\epsilon,ij} \ , \tag{11.14}$$

with

$$U_{\text{PSP}}(t) = \frac{I_S \ \cdot w_{ij}}{G - aC} \left(\mathrm{e}^{-a(t-\hat{t}_{\text{f},j})} - \mathrm{e}^{-(t-\hat{t}_{\text{f},j})G/C} \right) H(t - \hat{t}_{\text{f},j}) \tag{11.15}$$

and I_S the reference current.

Simplified IAF Model with Adaptive Synapses

As explained in chapter 2, we do not use a decay term for the membrane potential throughout this book. Hence, we further simplify the IAF model and set the membrane conductance equal to zero. A corresponding circuitry is presented in [67–69]. The output pulses last for $t_{\text{d}} = 1ms$, and the weights follow an adaptation rule inspired by Hebb's postulate (cf. (2.1c)). Weights, ie. synapses, and inputs are constrained to positive values only.

Neuron

The signal flow in the neuron is shown schematically in Figure 11.3. The neuron i is designated by a state variable $x_i \in \{0,1\}$ having two values and a continuous state variable $a_i \in \mathbb{R}$, which represents the membrane potential. x_i is also the output signal of the neuron.

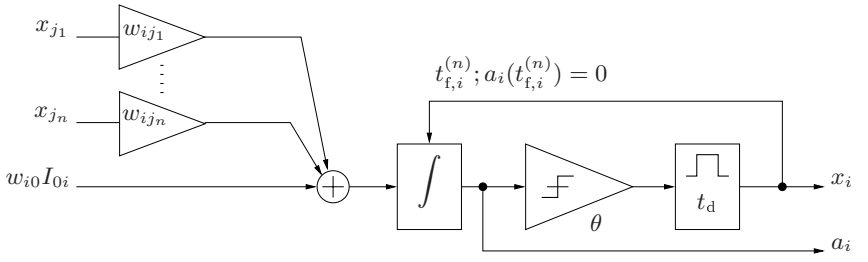


Fig. 11.3 Block schematic of the simplified IAF neuron and designation of signals

Between the pulse pauses we have $x_i = 0$, and the membrane is charged from zero to its threshold θ:

$$\left.\begin{aligned} \frac{da_i(t)}{dt} &= \sum_{j\in N_i} w_{ij}(t)\, x_j(t) + w_{i0} I_i \\ x_i &= 0 \end{aligned}\right\} \quad 0 \le a_i(t) \le \theta \ . \tag{11.16}$$

This is our original signal equation (2.1a). Besides the symbols $w_{ij}(t)$, already introduced for the adaptive synapses therein, I_i designates a continuous input to the neuron i which is weighted by w_{i0}. The index j runs through all neurons which are connected to the neuron i. Weights and inputs are positive only: $w_{ij}(t) \ge 0$, $w_{i0} I_i \ge 0$.

When the membrane potential a_i reaches the threshold θ the state of the neuron changes. This point of time is designated as n-th firing time $t_{\mathrm{f},i}^{(n)}$ of neuron $i : a_i(t_{\mathrm{f},i}^{(n)}) = \theta, \quad n \in \mathbb{N}$.

From then on the neuron fires: $x_i(t) = 1$, $a_i(t_{\mathrm{f},i}^{(n)} + t_\mathrm{d}) = 0$, $t_{\mathrm{f},i}^{(n)} \le t < t_{\mathrm{f},i}^{(n)} + t_\mathrm{d}$. After reaching its threshold, the neuron's state changes again, see above.

Because of the continuous input signal, the neuron acts as a pulse generator of frequency f provided there are no other inputs, ie. $\sum_{j\in N_i} w_{ij} x_j \equiv 0$. In this case f reads:

$$f_\mathrm{f} = \frac{1}{t_\mathrm{d} + \theta/(w_{i0} I_i)} \ , \tag{11.17}$$

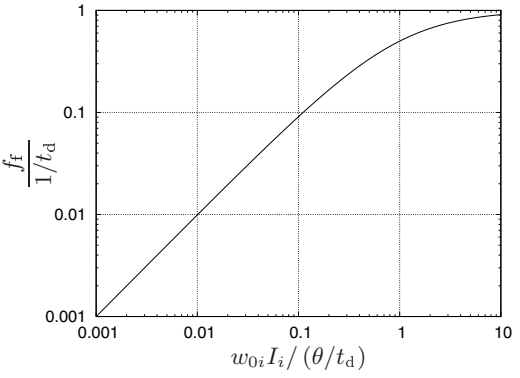


Fig. 11.4 Frequency f_f of the isolated IAF neuron as a function of the input $w_{i0}I_i$

the graph of which which is displayed in Figure 11.4 as a function of the input $w_{i0}I_i$.

Adaptive Synapses

An adaptive synapse is to increase or decrease its value such that the presynaptic neuron advances or lags behind the postsynaptic neuron. To this end, the continuous and discrete states of the coupled neurons are evaluated in a function unit, which is attributed to the synapse, in order to obtain the right phase relationship. Generally any of the locally available signals a_i, a_j and x_i, x_j can serve as one of these (Figure 11.5).

In [67] two adaptation rules were presented, one evaluating the two membrane potentials of the pre- and postsynaptic neuron, the other combining the pulse from the presynaptic neuron with the membrane potential of the postsynaptic neuron. As only the latter constitutes a local cause and effect relationship, we arrive at the synapse equation presented already in chapter 5:

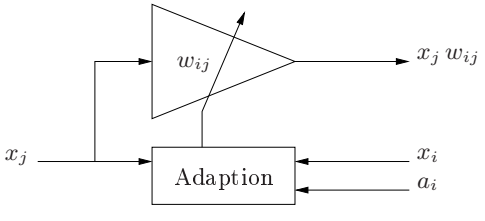


Fig. 11.5 Weight adaptation as a function of the local signals

$$\dot{w_{ij}}(t) = -\gamma \cdot w_{ij}(t) + \begin{cases} \mu \cdot (a_i(t) - \theta/2) & \text{if } x_i = 0 \wedge x_j = 1 \\ 0 & \text{otherwise .} \end{cases} \tag{11.18}$$

In chapter 12 the analogue implementation of this model and an adapted behavioural model are presented. For comparative studies such as synchronisation, acceptable tolerances and system simulation (chapter 13), the behavioural model is used.

Some authors have described biologically motivated circuits [70, 71]. The circuits of [70] contain inhibitory as well as excitatory synapses. Both Maass and Eckhorn [72] postulate that the extraction of correlations in input sequences of spikes - for example, in the V4 region of the mammalian vision complex for the purpose of feature detection - is one of the main tasks for these circuits. In this connection Eckhorn assumes that the correlation is realised as synchronous activity of the neurons and implemented by inhibitory synapses only. As was proven in chapter 5 the synchronisation of the neurons' pulse activity can only be realised with excitatory synapses and never with inhibitory ones.

Furthermore, our proposed mechanism of implementing correlation by means of excitatory synapses instead of lateral inhibitory ones seems to be plausible. In [73] measurements are presented suggesting that interactions between correlated receptive fields located in the V1 region of the visual cortex do not result from lateral inhibition, but from modulation of the vertical excitatory synapses.

12

Elementary Circuits for Neurons, Synapses, and Photosensors

Jörg Schreiter, Stephan Henker, and Jens-Uwe Schlüßler

12.1 Neuron

The membrane potential a_i of the IAF model (section 11.3) is realised as a voltage produced by the currents accumulating on the capacitor formed by the membrane of a neuron. A threshold switch with hysteresis determines the state variable with two values $x_i \in \{0, 1\}$ of neuron i by comparing the accumulator potential with a reference voltage. At the time of firing, x_i takes on the value 1 during the time span t_d, which is achieved by discharging the capacitor.

Circuit Concept for IAF Neuron

The voltage difference between the rest potential after firing and the threshold is limited to values of about 1 V by the supply voltage. With 10 nA, 1 V and 10 ms integration time (relating to a pulse rate of $100Hz$) a capacitor per MOS transistor of $100pF$ would result, which requires much too much area (about 320 mm^2 for a field of 128×128 neurons). Therefore, the integration time is reduced by a factor of 100, which is equivalent to a MOS capacitor of about $1pF$. The duration of a pulse was set to 1 μs.

According to (11.16) the membrane potential follows the integral of the input signals:

$$a_k(t) = \int_{t'=t_{\mathrm{f},k}^{(n)}+t_{\mathrm{d}}}^{t} dt' \left(\sum_{l \in N_k} w_{kl}(t')\, x_l(t') + w_{k0} I_k \right) . \qquad (12.1)$$

During the firing period ($x_k = 1$), the membrane potential will be discharged (see below). The gate-bulk capacitor of a MOS transistor is an optimal realisation of the membrane capacitor with regard to area. The charge and discharge currents of this MOS capacitor are handled by switched MOS transistors. Parasitic charge injections induced by the switching may cause a shift in the

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 195–231, 2010.
© Springer-Verlag Berlin Heidelberg 2010

membrane potential. The dimensioning of a circuitry which gives the optimal area thus demands special care. The main interrelationships are described in Appendix 19.3.

The membrane potential a_k is represented by a charge $Q(a_k)$ and a constant rest charge Q_0:

$$Q(a_k) = Q_0 + qa_k \ . \tag{12.2}$$

The states $a_k = 0$ and $a_k = \theta$ relate to the charges

$$Q_0 = Q(0) \quad \text{and} \quad Q_\theta = Q(\theta) \ . \tag{12.3}$$

The capacitance of integrated capacitors usually depends on the voltage and is given by $Q_C(U)$ or $C(U)$, with

$$C(U) = \frac{d}{dU} Q_C(U) \ . \tag{12.4}$$

The membrane potential a_k is thus mapped on to the voltage $U(a_k)$:

$$Q_C\big(U(a_k)\big) = Q(a_k) \ . \tag{12.5}$$

As for Q_0 and Q_θ, the states $a_k = 0$ and $a_k = \theta$ are fitted by the voltages U_0 and U_θ, with $Q_C(U_0) = Q(0)$ and $Q_C(U_\theta) = Q(\theta)$. Generally, however, it does not exist $\tilde{U}$ such that holds: $U(a_k) = U_0 + a_k\tilde{U}$, for all $a_k \in [0, \theta]$. Figure 12.1 shows charge $Q(a_k)$ and voltage $U(a_k)$ of the MOS transistor. The direction of the currents is related to the circuit chosen for the weights. With positive input signals, the voltage $U(a_k)$ declines.

For the realisation of the adaptation function $\mu \cdot (a_i(t) - \theta/2)$, the voltage $U(a_k)$ can be used provided the deviation from the linear capacitor is not too large. For a NMOS transistor this is the case when $U(a_k) \geq U_{\text{th}}$ and for PMOS when $U(a_k) \leq U_{\text{DD}} - U_{\text{th}}$. Here U_{th} designates the threshold voltage of the MOS transistor and U_{DD} the supply voltage. Already with a simple circuit,

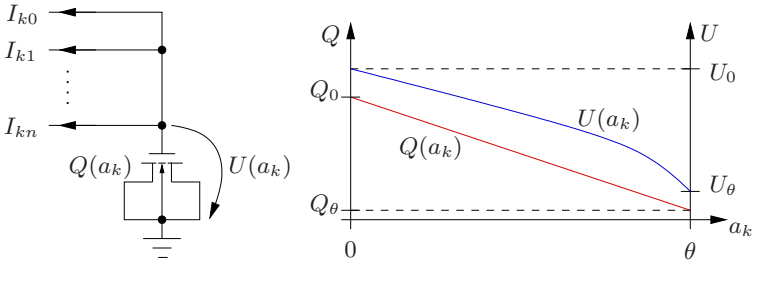


Fig. 12.1 Left: Integration of input signals by a MOS capacitor. Right: Exemplary mapping of membrane potential on to charge and voltage.

the hysteresis of the threshold switch can be extended to cover $|U(\theta)-U(0)| \approx U_{\mathrm{DD}} - U_{\mathrm{th}}$. Hence, the region of the MOS capacitor which is approximately linear can be used to its full extent.

The input signals are represented by currents I_{k0} ... I_{kn}:

$$I_{k0} = I_{\mathrm{N}} w_{k0} I_k \tag{12.6}$$

and

$$I_{kl}(t) = I_{\mathrm{N}}\, w_{kl}(t)\, x_k(t)\ , \tag{12.7}$$

with I_{N} designating a reference current and $l \in \{1, 2, \ldots, |N_k|\}$ indexing those neurons l connected to neuron k ($|N_k|$ indicating the number of such neurons).

The input currents $I_{k0}, \ldots I_{kn}$ discharge the accumulator, ie. the MOS capacitor, from U_0 to U_θ. In order to obtain the pulse signal and recharge the accumulator from U_θ to U_0, the circuit according to Figure 12.2 is used. The hysteresis of the switch is set such that it comes close to U_0 and U_θ, resp.

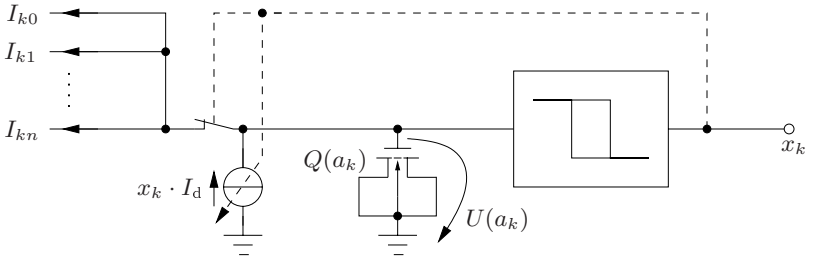


Fig. 12.2 Block diagram of a neuron

Once the membrane potential reaches the threshold for firing, ($a_k = \theta$, $U(a_k) = U_\theta$), the threshold switch outputs $x_k = 1$. Simultaneously the input current $I_i = I_{k0} + \ldots + I_{kn}$ is switched off and the charge current I_{d} switched on. Hence, the voltage $U(a_k)$ climbs to U_0, i.e. the membrane potential declines to zero. When reaching $a_k = 0$ or $U(a_k) = U_0$, the switch changes again and the integration of the input signals begins anew, while $x_k = 0$.

Principally, a Miller integrator helps reduce the area of the integration capacitor. A detailed investigation revealed that this benefit was outweighed by the need for the extra area for the inverting amplifier and the isolated capacitor.

Charge Current Source

The charge current for the accumulator is provided by a simple current source with a switch according to Figure 12.3b.

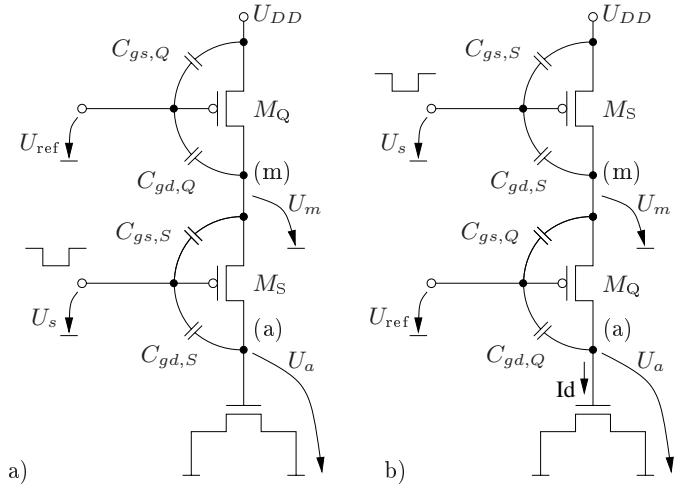


Fig. 12.3 Charging the membrane capacitor with the help of a current source and a switch. The current passing through the current source transistor M_{Q} is adjusted by means of a reference U_{ref} voltage. The current is switched on/off by the switch voltage U_s.

Neglecting for a moment the dependency of the output current I_{d} on the output voltage U_a, according to Figure 12.2, a simple relation between I_{d} and the pulse duration t_{d} results:

$$t_{\mathrm{d}} = (Q_0 - Q_\theta)/I_{\mathrm{d}} \ . \tag{12.8}$$

To first order holds:

$$\left|\frac{\Delta t_{\mathrm{d}}}{t_{\mathrm{d}}}\right| = \left|\frac{\Delta I_{\mathrm{d}}}{I_{\mathrm{d}}}\right| \ . \tag{12.9}$$

Similarly, this holds true for the charging of a non-linear capacitor with a voltage dependent current:

$$t_{\mathrm{d}} = \int_{U_\theta}^{U_0} \frac{C(U)}{I(U)}\, dU \ . \tag{12.10}$$

To first order it now holds:

$$\left|\frac{\Delta t_{\mathrm{d}}}{t_{\mathrm{d}}}\right| = \left|\frac{\Delta I_{\mathrm{d}}}{I_{\mathrm{d}}(U_*)}\right| \ , \quad \text{with } U_\theta \le U_* \le U_0 \ . \tag{12.11}$$

From the system simulations described in chapter 13 it can be concluded that a standard deviation of $\sigma(t_{\mathrm{d}}) \le 1\,\%$ can be tolerated provided the pulse

duration t_d follows a normal distribution. Because of (12.9) and (12.11), resp., the charge current I_d inherits the same relative deviation:

$$\frac{\sigma(I_\mathrm{d})}{I_\mathrm{d}} = \frac{\sigma(t_\mathrm{d})}{t_\mathrm{d}} \leq 0.01. \tag{12.12}$$

According to Appendix 19.3 the local random relative deviation of the drain current is largest when operated in weak inversion. The relation (12.12) is fulfilled safely and independently from the selected charge current when it holds for weak inversion.

From the mismatch parameters of the CMOS manufacturing technology it follows a minimal gate area of $9\mu\mathrm{m}^2$ - assuming L$> 0,8\mu$m - in order that $\sigma(t_\mathrm{d})/t_\mathrm{d} \leq$ 0,01 holds. With $W = L = 3\mu$m, also the assumption on the minimal gate length holds.

The switch transistor is placed at the source side of the current source transistor (Figure 12.3b). Compared to Figure 12.3a, when switching the current source *on* or off, the disturbance of the membrane voltage U_a is minor.

Threshold Switch

As mentioned earlier, the threshold switch should have a large hysteresis in order to exploit the voltage range defined by the supply voltage to its full extent and minimise the area of the integration capacitor. The realised threshold of the neuron is proportional to the hysteresis and therefore displaceable externally. System simulations (chapter 13) recommended a standard deviation at an accepted maximum level of less than 1 *percent* for the threshold. This holds also for the hysteresis. The requirements are fulfilled with a threshold switch realised as differential amplifier with positive feedback (Figure 12.4). So the output signal, which is fed back to the non-inverting input, is limited by the upper or lower value of the threshold of the switch.

To explain the function a hard clipping is assumed:

$$U_\mathrm{fb} = \min\left(\max\left(v \cdot U_d + \frac{U_\mathrm{plus} + U_\mathrm{minus}}{2},\ U_\mathrm{minus}\right),\ U_\mathrm{plus}\right), \tag{12.13}$$

where $v \gg 1$ designates the differential gain, U_minus and U_plus the lower or upper threshold.

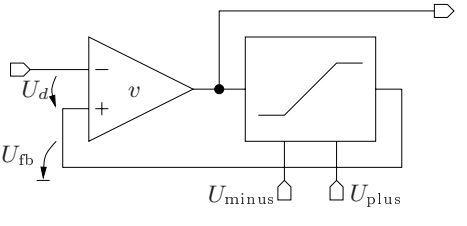


Fig. 12.4 Functional principle of the threshold switch

For input voltages between the thresholds $U_{\text{minus}} + \Delta U < U_a < U_{\text{plus}} - \Delta U$ there exist two stable values for the threshold: $U_{\text{fb}} = U_{\text{plus}}$ and $U_{\text{fb}} = U_{\text{minus}}$. From $U_{\text{fb}} = U_{\text{plus}}$ the threshold switch turns to $U_{\text{fb}} = U_{\text{minus}}$ if $U_a > U_{\text{plus}} - \Delta U$. Accordingly, it switches from $U_{\text{fb}} = U_{\text{minus}}$ to $U_{\text{fb}} = U_{\text{plus}}$ if $U_a < U_{\text{minus}} + \Delta U$. It results from (12.13): $\Delta U = (U_{\text{plus}} + U_{\text{minus}})/2v$.

The threshold values are provided as reference voltages. The input offset voltage U_{offs} of the amplifier in the threshold switch, which leads to an offset $U'_a = U_a + U_{\text{offs}}$ of the input voltage, acts on the upper and lower threshold values in an equal manner, and hence is compensated fully. It results in an almost constant threshold θ for all neurons.

The circuit of the soma with integration capacitor and accumulator, resp., threshold switch, and switch for the input current is shown in Figure 12.5. The threshold switch is implemented as a 3-stage amplifier with positive feedback which is limited by the switches M_{P3} and M_{N4}.

The first stage is composed of a differential pair (M_{N1}, M_{N2}) with current mirror (M_{P0}, M_{P1}). The tail current of the differential pair (through M_{N0}) flows continuously. Because of power consumption, it must be chosen as small as possible. On the other hand, when diminishing the tail current, the switching velocity of the whole circuit gets reduced. This implies a delay between the input signal, when passing the threshold, and the rise of the output signal. Furthermore, the circuit stays on longer between the stable states, hence a considerable cross current flows in the output stages of the threshold switch. A favorable compromise is achieved with a tail current of 20 nA.

The second stage is a simple PMOS source stage M_{P7} with source current load M_{N13}, which considerably contributes to the voltage gain of the amplifier in the threshold switch. Its current is fixed for the same reasons as was the tail current of the differential amplifier. For the sake of simplicity, both current source transistors are dimensioned equal.

The third stage is a simple inverter dimensioned minimally. At its input, the input signal is amplified already to such a level that the voltage range causing the cross current is passed sufficiently fast.

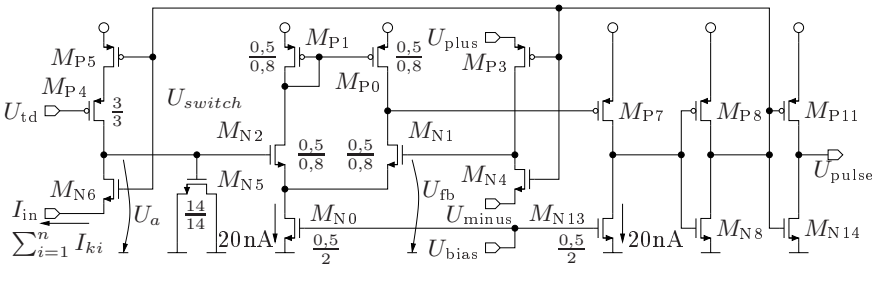


Fig. 12.5 Circuit of the soma: W/L relations of the transistors are specified in $\frac{1\mu\text{m}}{1\mu\text{m}}$, otherwise they are minimal: W/L=0,4/0,5.

The positive feedback is realised with the switch M_{N4}-M_{P3} by which, depending on the switching state, the upper (U_{plus}) or lower (U_{minus}) threshold value is passed to the non-inverting input of the amplifier.

The circuit is complemented by the output inverter M_{P11}-M_{N14}, the discharge current source M_{P4}-M_{P5}, and the switch transistor M_{N6}. The output inverter uncouples the „high“ signal of the neural pulses from the remaining neurons of the net. The switching transistor and the disengageable discharge current source serve to switch between the states „accumulation $x = 0$“ and „firing $x = 1$“ of the neuron.

Dimensioning

Usually, for transistors used in analogue circuits $l > l_{\mathrm{min}}$ is chosen. Furthermore, the transistors of the differential amplifier M_{N1}, M_{N2}, M_{P0} and M_{P1} should match closely. The description of matching in the manufacturing technology under consideration warns that the matching of lengths below $2 \cdot l_{\mathrm{min}}$ declines sharply. For the width of the transistors such a restriction does not exist. Thus the dimensioning for the 4 transistors was chosen to read: $W/L = w_{\mathrm{min}}/2\,l_{\mathrm{min}} = 500\,\mathrm{nm}/800\,\mathrm{nm}$. For most of the other transistors it sufficed to choose $W/L = w_{\mathrm{min}}/l_{\mathrm{min}} = 500\,\mathrm{nm}/400\,\mathrm{nm}$ in order to achieve a sufficiently fast switching velocity.

Only the lengths of the transistors M_{N0} and M_{N13} used for the current source are chosen such that the influence of the output resistance is negligible: $W/L = w_{\mathrm{min}}/5\,l_{\mathrm{min}} = 500\,\mathrm{nm}/2\mu\mathrm{m}$.

A particular task of dimensioning concerns the duration of the switching event, for it may not influence the duration of the pulse. Therefore the difference between the duration of the switching at the start of the pulse (from $U_{\mathrm{fb}} = U_{\mathrm{minus}}$ to$U_{\mathrm{fb}} = U_{\mathrm{plus}}$) and the one at the end of the pulse (from $U_{\mathrm{fb}} = U_{\mathrm{plus}}$ to $U_{\mathrm{fb}} = U_{\mathrm{minus}}$) may not vary too much from neuron to neuron. This can be achieved in the simplest manner by keeping the duration of the switching process sufficiently small[1]: $\max(t_r, t_f) \leq t_{\mathrm{d}}/100$.

Dynamic Behaviour

When the neuron receives the input signals $I_{k0} \ldots I_{kn}$, pulses will be generated at intervals of $t_I = (Q_0 - Q_\theta)/I_{Sk}$ provided $I_{Sk} = \sum_{i=1}^{n} I_{ki} > 0$. For $I_{Sk} \leq 0$ the neuron remains in the state $x = 0$. For $I_{Sk} < 0$ the voltage U_a increases until the parasitic diode of the drain region of M_{P4} in Figure 12.5 becomes conductive and bleeds off the input current I_{Sk}[2].

[1] It is to be remembered that the relative standard deviation of the pulse duration t_{d} must be below 1 *percent*.

[2] In this book only the case $I_{Sk} > 0$ is to be found. Note that even for switched-off inputs the output transistors of the adaptive synapses (section 12.2) and of the continuous input (section 12.4) provide a weak inversion current such that $I_{Sk} > 0$.

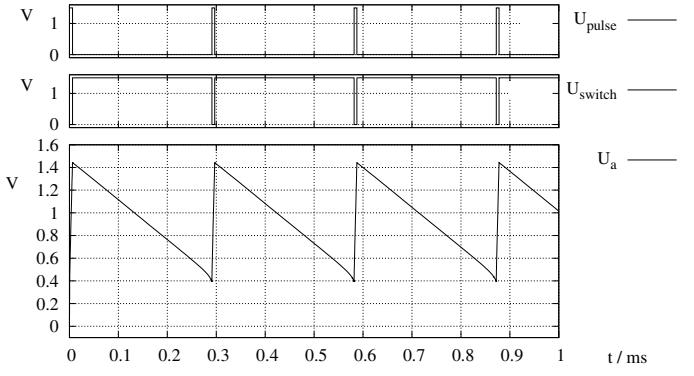


Fig. 12.6 Pulse generation in the neuron for constant input $\sum_{i=1}^{n} I_{ki} = 3,5\,\mathrm{nA}$.

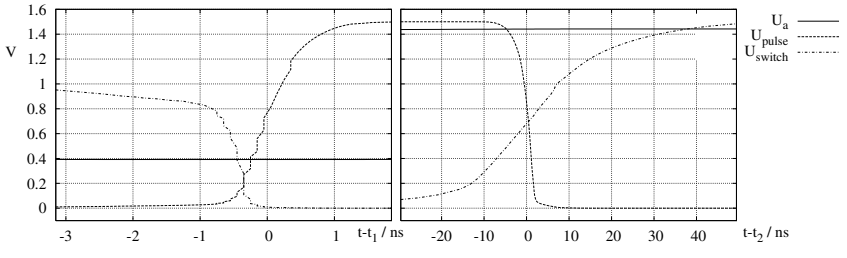


Fig. 12.7 Details of the signal response from Figure 12.6 at the switching points $U_{\mathrm{pulse}}(t_1) = U_{DD}/2$, $U_a(t_1) \approx U_{\mathrm{minus}}$ and $U_{\mathrm{pulse}}(t_2) = U_{DD}/2$, $U_a(t_2) \approx U_{\mathrm{plus}}$.

Figure 12.6 displays the response of the voltage U_a and the pulse output U_{pulse} for a constant input signal. The response is simulated for the circuit in Figure 12.5.

In order to describe the behaviour at the switching points (Figure 12.7) the threshold switch must be examined, together with its input signal I_{Sk} and the switch between it and the discharge current source M_{N6}-M_{P5}.

During the integration phase $U_{\mathrm{fb}} = U_{\mathrm{minus}}$ holds, and the input current $I_a = I_{Sk}$ flows out of the integration capacitor. The membrane voltage U_a follows the integral of the input signals: $U_a(t) = U_{\mathrm{plus}} - \int_{t_f}^{t} I_a(t')\,dt'$. When the membrane voltage U_a reaches the lower threshold value U_{minus}, U_{fb} increases, and because of the positive feedback, the neuron starts to fire a pulse. At the same time the input current of the integrator jumps from $I_a = I_{Sk}$ to $I_a = I_{\mathrm{d}}$. During firing $U_{\mathrm{fb}} = U_{\mathrm{plus}}$ holds, and the membrane voltage increases until U_{plus} is reached and the neuron switches back to integration.

Besides the positive feedback caused by the switching of the threshold with the transistors M_{P3} and M_{N4}, there is some negative feedback caused by the

switching between the input signal and the discharge current source realised by the transistors M_{P5} and M_{N6}. It must be guaranteed that the positive feedback prevails at the switching points and the neuron neither reaches a stable operating point nor falls into an oscillation around the lower switching point.

A stable operating point would assume the existence of a fixpoint in the system:

$$I_a = 0 \tag{12.14}$$

$$U_a = U_{\mathrm{fb}}. \tag{12.15}$$

Equation (12.14) holds when the switch M_{N6}-M_{P5} between the input current $\sum_{i=1}^{n} I_{ki}$ and the discharge current I_{d} stands in an intermediate position. Because of the integration capacitor U_a is almost unchanged. Therefore (12.15) holds when $U_{\mathrm{fb}} \approx U_{\mathrm{minus}}$ and $U_{\mathrm{fb}} \approx U_{\mathrm{plus}}$, respectively.

The corresponding common gate voltage U_{switch} of the transistors M_{P3}, M_{N4}, M_{P5} and M_{N6} can be estimated as follows. The input current $\sum_{i=1}^{n} I_{ki}$ lies in the weak inversion range of the threshold transistor M_{P5} of the discharge current source. In order that (12.14) holds, U_{switch} must range between U_{DD} and $U_{DD} - U_{\mathrm{th}}$. Then M_{N4} is fully on provided U_{minus} is chosen not too high. M_{P3} runs a weak inversion current flowing through M_{N4}, thereby causing a very low drain source voltage. Thus $U_{\mathrm{fb}} \approx U_{\mathrm{minus}}$ holds, meaning that a fixpoint can only exist at the lower switching point.

In the present circuit those operating points lying near the lower switching point are as unstable as the fixpoint in the DC voltage analysis. This is a result of the small signal analysis, which is presented in Figure 12.8.

The three parts of the diagram A show a section of Figure 12.6. The arrow designates the lower switching point $U_a \approx U_{\mathrm{minus}}$. In the diagrams B1 and B2 the accumulator voltage U_a is displayed near the switching point, as well as the positive feedback voltage U_{fb} and the accumulator current I_a. The zero point on the time axis is shifted to the place $U_a(t_0) = U_{\mathrm{fb}}(t_0)$. At selected places (E - L) small signal analysis was performed.

The input signal is represented by a small signal current source at the accumulator (ie. parallel to U_a), and the output signal is the switch voltage U_{switch}. The small signal analysis for various operating points is shown in the diagrams D to L as frequency response of the transfer impedance $\underline{Z}(p) := \underline{U}_{\mathrm{switch}}(p)/\underline{I}_a(p)$ using non-linear coordinates. The modulus of $\underline{A}(p)$ is given as $|\underline{A}(p)| := k \log(|\underline{Z}(p)|/Z_0)$ with suitably chosen values for k and Z_0 whereas the argument is kept: $\arg \underline{A}(p) := \arg \underline{Z}(p)$. Frequencies range up to $10GHz$, and decades are marked by points along the curves. The point with $f \approx 0$ is accented by a small circle. The diagram D shows the small signal analysis for the unstable point of equilibrium obtained for the circuit in the DC voltage analysis. The other small signal analysis was performed at the operating points mentioned above. A detailed study shows that the apex in diagrams D to H is caused by two poles in the right half of the complex frequency plane p.

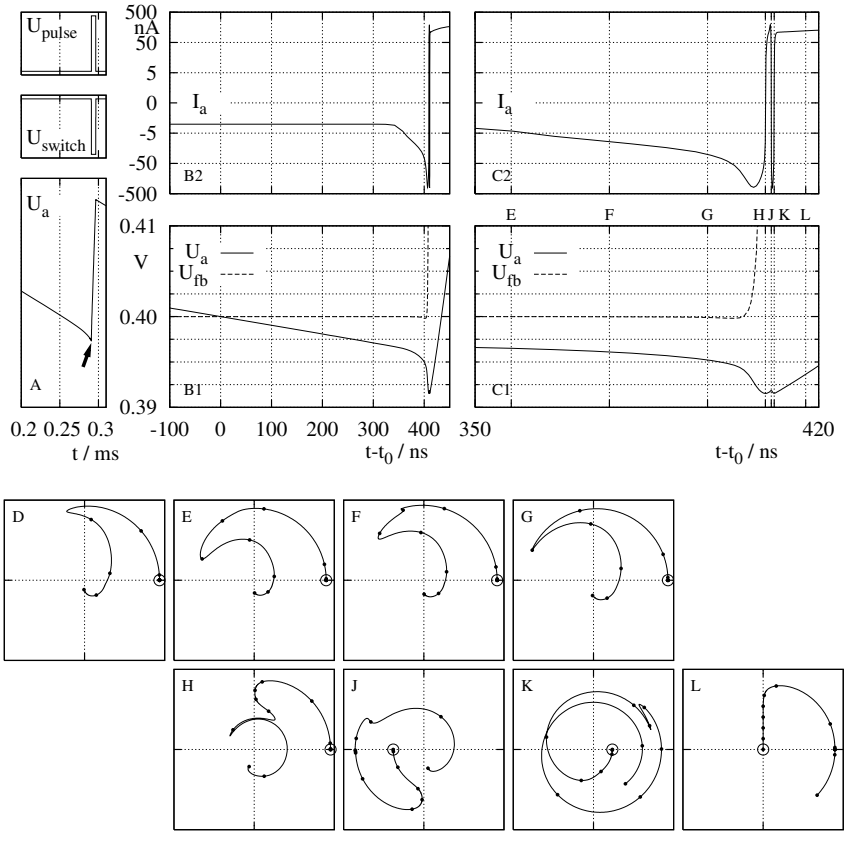


Fig. 12.8 Small signal behaviour of the soma at the point of equilibrium and in the neighbourhood of the lower switching point

The instability of the operating points can also be read off from the frequency responses in the quadrants I, II, and IV. In conclusion, after the accumulator voltage U_a has fallen slightly below the lower threshold value, the soma stays at an unstable operating point, which leads to switching. Not until the threshold value has been considerably changed via positive feedback will the circuit run through a sequence of locally stable operating points, as can be seen in diagrams J to L.

It is obvious that the switching speed is sufficiently large in the useful input signal range. The duration of the switching and the rise and fall time of the output signal are well below one hundredth of the pulse duration of 1 μs.

In summary, the results show that the requirements for the soma circuit are fulfilled with minimal dimensions for the transistors and empirically obtained values for the length of the analogue transistors, and their currents at the operating points.

Layout

The preceding reflections on the optimal dimensioning of circuitry were based on analytical calculations and network simulations. They have resulted in the right values of parameters for the geometric dimensions of transistors and capacitors. The challenge, then, is to construct a neuron cell with a minimal area which can be used about 10000 times in the chip. In Figure 12.9 the complete layout of the neuron cell is shown, comprising synapse circuitry as well as read-out and test circuits.

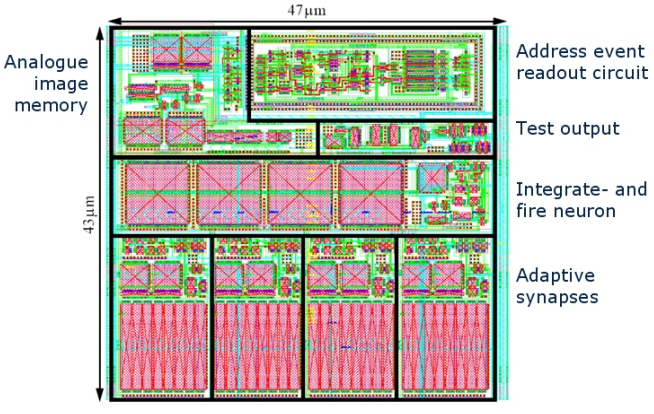


Fig. 12.9 Layout of a neuron circuit with 4 adaptive synapses as well as read-out and test circuits

12.2 Adaptive Synapses

The block diagram of the synapse model is shown in Figure 12.10. The accumulator and threshold circuit in the right part of the figure has four synapses. Each one is comprised of the memory for the synaptic weights w_{kl} and an adaptation circuit, which implements the adaptation rule (11.18) obeyed by the synapses.

The voltage across the MOS capacitor sinks from the upper switch point of the Schmitt-Trigger at $a_k = 0$ to the lower point at $a_k = \theta$. The circuit for the synapses shall possess the following properties:

- analogue storage of the synaptic weights w_{ij},
- continuous change of the weight in proportion to the input signal: $I_{\text{adapt}} \propto \dot{w}_{ij}$,
- output current proportional to the synaptic weight and switched by the neural output signal x_j: $I_{\text{pulse}} \propto w_{ij} x_j$.

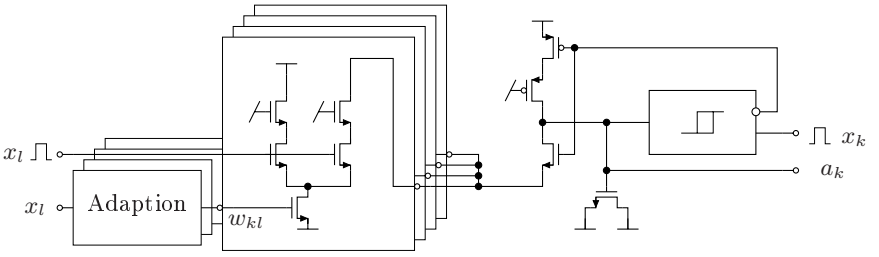


Fig. 12.10 Block diagram of a neuron with four synapses

Storage of Synaptic Weights

The particular simple realisation of Figure 12.11 arises from the synaptic weight w_{ij} being the gate charge of a MOS transistor: $Q_G = Q_0 w_{ij}$. This charge can be varied continuously through the input current I_{adapt}: $\dot{w}_{ij} = \dot{Q}_G/Q_0 = I_{\text{adapt}}/Q_0$.

The drain current of the transistor is switched on or off, and serves as output current. So that it is approximately proportional to the gate charge, the transistor is operated in the Ohmic range, similar to the current-voltage converter of Vittoz ([74]). As well as that, the drain voltage of M_{N0} is kept almost constant with the help of the transistor M_{N5}. Its gate-source voltage changes at $65mV$ per decade of the drain current. The useful voltage range

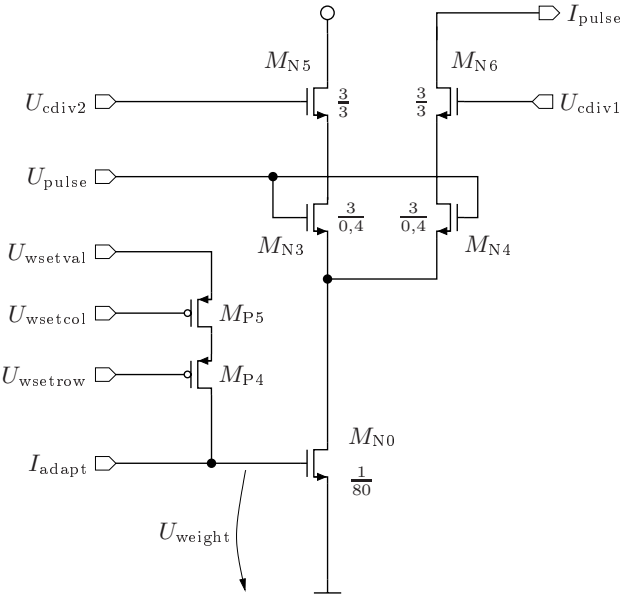


Fig. 12.11 A synapse's circuit

(U_{weight}) at the gate of the weight storage transistor lies between 0,4 V - the threshold voltage of M_{N0} - and 1,4 V, thus about 0,1 V below U_{DD} because of the saturation of the circuit which provides the adaptation current I_{adapt}. The range for the weight current and for the adaptation current for the continuous adaptation according to (11.18), resp., was fixed at $I_{\text{pulse}} = (0 \ldots 2)\,\text{nA}$ and $I_{\text{adapt}} = (-2 \ldots 2)\,\text{nA}$, resp. The maximum change rate of the weight amounts to $\max(|\dot{w}_{ij}|) = 0,01$. In the system simulation weight values up to $w_{ij} \leq 0.2$ were obtained. In order to utilize the weight storage close to optimal, the maximum weight should equal the useable voltage range of about 1 V. Then, the gate capacitor for the weight storage transistor reads:

$$\begin{aligned} C_{G,M_{\text{N0}}} &= \frac{\max(I_{\text{adapt}})\, t_{\text{d}}\, \max(|w_{ij}|)}{1V\ \max(|\dot{w}_{ij}|)} \\ &= \frac{2\,\text{nA} \cdot 10\,\mu\text{s} \cdot 0,2}{1V \cdot 0,01} \\ &= 400\,\text{fF}. \end{aligned} \tag{12.16}$$

This requires a gate area of $W \cdot L = 80\,\mu\text{m}^2$. In order to achieve approximate linearity with regard to voltage-current conversion, M_{N0} must be operated in the Ohmic range and in strong inversion. This is only accomplished with a drain current of 100 nA. Hence, a current divider comprised of transistors $M_{\text{N5}}$and M_{N6} is inserted in order to get the small output current range of $0 \ldots 2\,\text{nA}$. We have:

$$\frac{I_{D6}}{I_{D5}} = e^{\frac{U_{\text{cdiv1}} - U_{\text{cdiv2}}}{nU_T}}\ . \tag{12.17}$$

For $I_{D6}/I_{D5} \approx 100$ the W/L ratio of M_{N0} reads $W/L = 1\mu\text{m}/80\mu\text{m}$. In the layout M_{N0} is replaced by a connection in series of 8 transistors each with $W/L = 1\mu\text{m}/10\mu\text{m}$, which will not change the action of the circuit, however.

The drain current of the whole circuit is switched off by the switch transistors M_{N3} and M_{N4}. Note that they can not be replaced by a single switch transistor connected in series with M_{N0} because a current path from U_{DD} to I_{pulse} would exist otherwise in switch off mode.

The circuit is complemented by two select transistors M_{P4} and M_{P5} which preselect the weight values by means of an externally applied voltage U_{wsetval}.

Figure 12.12 displays the course of the voltage U_{weight} at the weight storage and the output current I_{pulse} when the synapse is switched on, depending on the charge Q_{weight} which represents the weight value. The fairly large deviation from the linearity of $I_{\text{pulse}} \propto Q_{\text{weight}} - Q_0$ does not disturb the synchronisation and is accepted because of the simplicity of the circuit design.

The adaptation processes run so slowly that the synapse circuit can be operated semi-statically and its dynamic behaviour neglected, except for the integration of the adaptation current I_{adapt}. The disruption of the membrane voltage U_a of the subsequent neuron, when switched on or off, can be kept low by separating the switch transistor M_{N4} from the output by means of the current divider transistor M_{N6}.

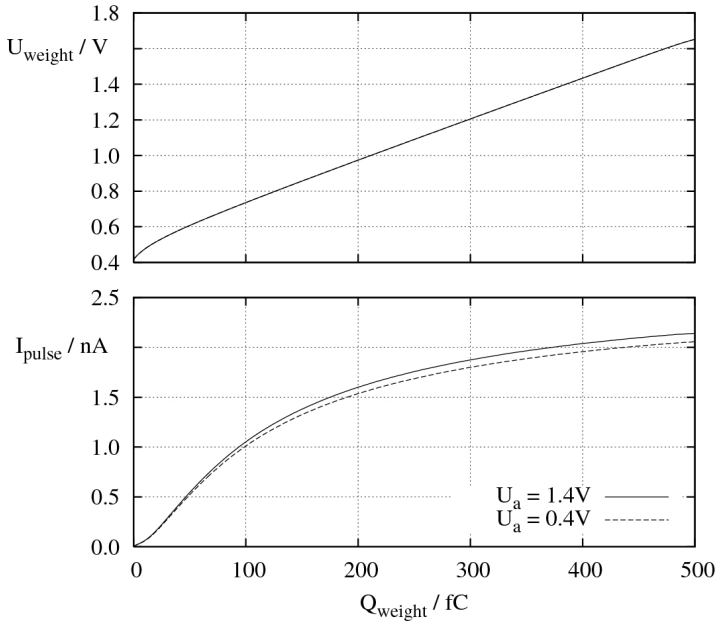


Fig. 12.12 Voltage at the weight storage and synapse's output current for $U_{\text{pulse}} = U_{DD}$

Adaptation of Synapses

As mentioned earlier, the adaptation rule needs to be implemented approximately only, and in consequence the respective circuit can be designed simply, power-efficiently and small in size. According to (11.18) the adaptation rule reads:

$$\dot{w_{ij}} = -\gamma \cdot w_{ij} + \mu \cdot (a_i - \theta/2) \ . \tag{12.18}$$

The product $\mu \cdot (a_i - \theta/2)$ is realised by a two-quadrant multiplier comprising a differential transistor pair and a tail current source. At the tail current source the factor μ is regulated. At the inputs of the differential pair the voltage U_a of the membrane potential and the one for the fixed value of the threshold are applied. The entire adaptation circuit is shown in Figure 12.13.

The differential equation realised by this circuit reads:

$$\dot{w_{ij}} = \begin{cases} -\gamma \cdot w_{ij} + \mu \tanh(\frac{a_i - \theta_i/2}{v_i}) & t_j < t < t_j + t_{d,j} \\ -\gamma \cdot w_{ij} & \text{otherwise} \ , \end{cases} \tag{12.19}$$

with v_i a normalisation constant. The term $\gamma \cdot w_{ij}$ is replaced by the function I_1 (Figure 12.14 top) and realised by the transistor M_γ which has a fixed gate source voltage.

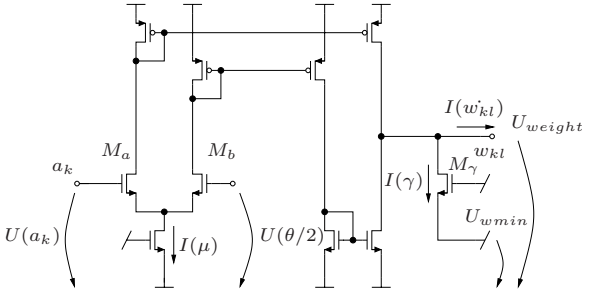


Fig. 12.13 Circuit for adaptation

In order to allow for sufficiently large voltage ranges of the input signal $U_{a,i}$ as well as for the output voltage U_{weight}, the output signal must be conflated by means of two current mirrors.

The system simulation showed that despite strong simplifying assumptions, the synchronisation of the neurons could not be compromised. The transistor M_γ can be operated such that it becomes saturated at very small voltages $U_{\text{weight}} - U_{\text{wmin}}$ which correspond with the very small weights. As

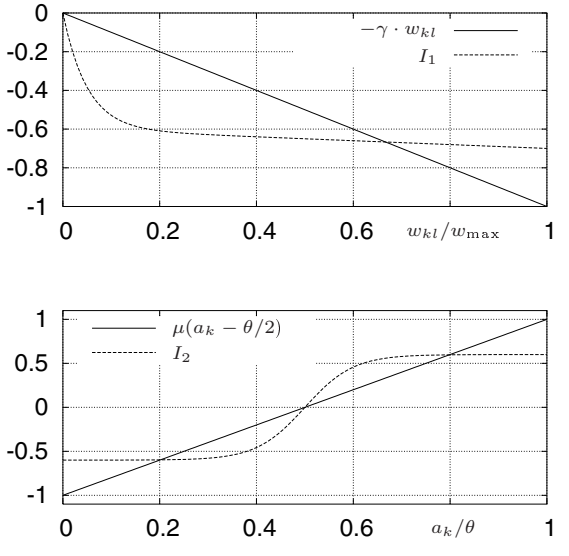


Fig. 12.14 Comparison of idealised adaptation function with the one implemented for the case of the terms γw_{ij} (top) and $\mu \cdot (a_i - \theta/2$ (bottom). The adaptation current is proportional to the weight change and is composed of two currents, $\dot{w_{kl}} \sim I(\dot{w_{kl}}) = I_1 + I_2$.

only positive weights are used, its behaviour for $U_{\text{weight}} < U_{\text{wmin}}$ has not to be considered. Similarly, the input voltage of the differential pair M_{a} and M_{b} of the multiplier should not be restricted to the approximately linear range $|U_{a,i} - U_\theta/2| < 2nU_T$. The voltage U_{weight} can be applied directly.

In summary, the implementation shown in Figure 12.13 represents quite a rough approximation of the adaptation rule (11.18). Nevertheless, this is acceptable since it performs the desired synchronisation.

12.3 Photosensors

This section proposes circuits necessary to stimulate neural nets with visual information. The image sensor is defined to be a system for spatially resolving measurements of the radiated power of visible light, which serves for image recording. Image sensors are usually composed of a 2D array of single photo detectors which are also designated as pixels. A real image is cast at the image sensor, the photo detectors of which proportionally convert the incident light into electrical signals. As the number of available signal paths for reading off the photo detectors is often limited, the pixel of the 2D array are addressed serially. The analogue signal value is digitised and stored as a matrix, thus representing the grey or colour values of the pixels. For our purposes, the signal values of the single pixels can be directly used as input to the neural net under consideration. In doing so, either the output current or voltage of the pixel or the respective pulse rate can be selected to carry the information.

The Photoelectric Effect

Visible light between 380 and $700nm$ is not directly measurable. Besides photo-chemical methods there are also electrical ones. In contrast to the plethora of chemical reactions triggered by incident light, only one physical effect exists which relates light and electrically measurable quantities. The photoelectric effect describes a process in which light transfers its energy to electrons. In case of the outer photoelectrical effect, electrons step outside of the surface of the illuminated metal. The first observations came from Heinrich Hertz in the year 1887, and then in 1905 Albert Einstein could explain the photo effect with the quantum theory of light. Accordingly, a light quant of frequency f possesses the energy $E = hf$, with h Plank's constant. In case of the outer photo effect, an electron of mass m_e steps out of the material, provided the transferred energy is larger than the work function W_A. It holds:

$$h \cdot f = W_a + \frac{m_e \cdot v^2}{2} \ , \tag{12.20}$$

with v the velocity of the electron after leaving the metal.

Semiconductive Photosensors

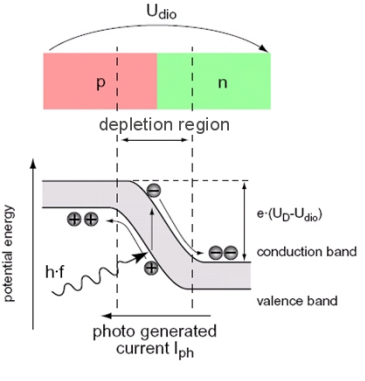
In the case of semiconductors, the internal photo effect comes to the fore. The mechanism is the same for both effects, but here electrons do not leave the semiconductor, but change from the valence band to the conduction band provided the transferred energy is sufficient to skip the band gap. The internal photo effect forms the basis for modern radiation measurement, photovoltaics, and the technology of digital cameras.

Photosensors made of semiconductors are particularly simple to integrate into existing microelectronic processes. Compared to other systems they excel with higher sensitivity and dramatically reduced power consumption. They are easily interfacing with digital circuits, since only an analogue-digital converter is needed.

The analysis of the opto-electronic properties of semiconductive photosensors relies on knowledge of the spectral optic generation rate $G(\lambda, x)$, which gives the number of generated charge carriers to a depth of x per second as a function of the wavelength λ. A detailed survey of the context of the optical generation is found in Appendix 19.4.

Photodiode

Beside the photoresistor, the photodiode is the most common electronic device for measuring the radiation of light. It is built as a multi-layered semiconductor structure, thus featuring several potential barriers. In all semiconductor technologies multi-layered structures are available for the construction of photodiodes. For special applications additional technological process steps may be needed, however.



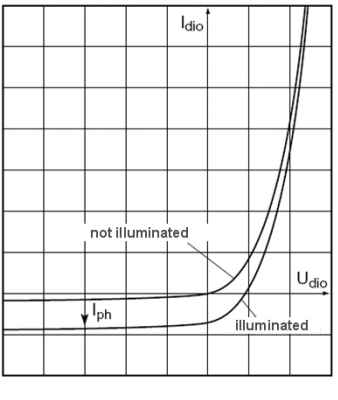


Fig. 12.15 Semiconductor mechanism when illuminated, and $U - I$ characteristic of a photodiode

Because of the electrical field in the space charge zones of a multi-layered structure (pn junction), the charge carriers generated by the light separate and drift to opposite sites. This photocurrent becomes noticeable as a shift of the $U - I$ characteristic of the diode (Figure 12.15). Note that in the case of photovoltaics, the photodiode is operated in the 4th quadrant, whereas for the measurement of light radiation it is operated in the 3rd quadrant, ie. in reverse-biasing. With no illumination, but voltage applied, a very low current, the dark current I_0, is still flowing. With illumination, the photocurrent I_{ph} must be added.

For the determination of the spectral photocurrent density $j_{ph}(\lambda)$ the system of equations for the semiconductor needed to be solved taking into account the photogeneration. A common approximation assumes that all generated charge carriers add to the current flow. Then, the spectral photocurrent density equals the integral of the generation rate $G(\lambda, x)$ over the respective thickness of the active photoelectric layer of the semiconductor:

$$j_{ph}(\lambda) = e \cdot \int_{x_0}^{(x_0+d)} G(\lambda, x)dx = e \cdot \eta_q(\lambda) \cdot \pi(\lambda, 0) \cdot (1 - e^{-\beta(\lambda)\cdot d}) \cdot e^{-\beta(\lambda)\cdot x_0} \ . \tag{12.21}$$

The total density of the photocurrent is given by integrating over the wavelength. For this, the photonic flow density $\pi(\lambda, 0)$ of the incident radiation as well as the substance-related wavelength dependencies of $\eta_q(\lambda)$ and $\beta(\lambda)$ must be known. The thickness is generally given as that region of the respective device that possesses a sufficiently large field strength to separate the charge carriers. For an abrupt junction profile, the reverse-biasing thickness d_s reads according to [75]:

$$d_s = \sqrt{\frac{2\epsilon_H(N_A + N_D) \cdot (U_D - U_{dio})}{eN_AN_D}} \quad \text{with} \quad U_D = U_T \cdot \ln \frac{N_AN_D}{n_i^2} \ . \tag{12.22}$$

From equation (12.22) it follows that the photocurrent intensifies with the increasing thickness of the space charge zone. In order to avoid recombination of the generated charge carriers, their drift length $l_D = \mu\tau E$ must exceed the thickness d. In turn, the drift length is determined by the mobility μ of the respective charge carriers, their lifetime τ, and the electrical field strength E.

Furthermore, we see that the spectral sensitivity of a photodiode depends on the depth x_0 of the space charge zone. Depth and thickness are adjustable by means of the doping profile and, to some extent, by the reverse voltage. For instance, a photodiode can be realised as a well-substrate diode at a great depth or as a diffusion diode close to the substrate's surface. Also a combination of various diodes is possible. Dual-junction diodes possess two pn-junctions, triple-junction photodiodes have three stacked pn-junctions. However, such multi-layered photodiodes must meet the requirement that they

have equal absolute sensitivities and differ mainly in their spectral response. If the photodiodes are implanted at different depths, from equations (12.21) and (12.22) it follows that the diodes must exhibit similar reverse thickness and thus, similar doping concentrations in order to fulfil this requirement. Generally, the photonic density reduces with increasing depth, thus a reduced sensitivity is to be expected for radiation with larger wavelengths at greater depths. In most environments this is compensated for by means of the larger intensity for red and infrared radiation. Though the doping concentrations are restricted technologically and thus can not be used for the design, the knowledge of doping concentrations still indicates how suitable a technology under consideration is for the realisation of a multi-layered photodiode and respective sensors.

The photocurrents of photodiodes are generally low. For an illumination strength of $500\,lx$, which the human eye judges to be very bright, we get about $2{,}05{\cdot}10^6$ photons of wavelength $555\,nm$ incident on an area of $1\,\mu m^2$ per second. With a typical absorption coefficient β_{555nm} of $5{\cdot}10^3\,cm^{-1} = 0{,}5\,\mu m^{-1}$ for silicon a number of $1{,}03 \cdot 10^6$ generated electron-hole pairs results per μm^3 and second, assuming a quantum efficiency of 100%. Neglecting recombination and assuming all electrons contribute to the current flow in the photoactive volume of $10 \times 10 \times 1 \mu m^3$, a photocurrent of about $16\,pA$ results.

Integrating Photosensor

Photodiodes can be used to construct an integrating photosensor. The photoelectrons are accumulated on the reverse capacitor of the photodiode. The reverse capacitor depends on the voltage across the diode, and for an abrupt pn-junction with area A (neglecting fringe effects) it amounts to:

$$C_{sp} = A \cdot \sqrt{\frac{\epsilon e N_A N_D}{2(N_A + N_D)U_D}} \cdot \sqrt{\frac{U_D}{U_D - U}} = C_0 \cdot \sqrt{\frac{U_D}{U_D - U}} \ ,$$
$$\text{with } U_D = U_T \cdot \ln \frac{N_A N_D}{n_i^2} \ . \tag{12.23}$$

For constant photocurrent I_{ph} the voltage response at different times across the diode is controlled by the differential equation

$$C_{sp}(U) \cdot \frac{dU(t)}{dt} = -I_{ph} \ , \tag{12.24}$$

which has the solution:

$$U(t) = U(0) - \frac{(I_{ph}t)^2}{4U_D C_0^2} + \frac{I_{ph}t \cdot \sqrt{U_D^2 - U_D U(0)}}{U_D C_0} \ . \tag{12.25}$$

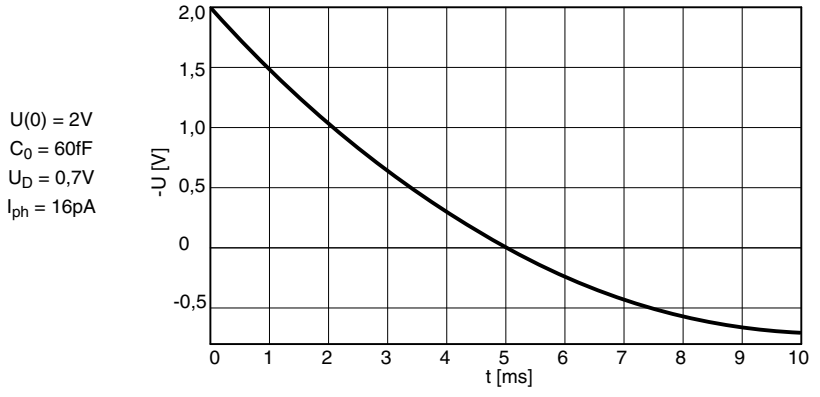


Fig. 12.16 Time-varying response of the diode voltage according to (12.24)

The response of the diode voltage (Figure 12.16) is in first order approximately linear for sufficiently short times. Thus, a linear relation between radiation strength and measurable voltage value results. Note that the photocurrent I_{ph} as well as the capacitor C are proportional to the diode area A. It follows that for fixed illumination strength the voltage across the diode does not depend from the area of the diode. This is correct in first order only, as a photodiode exhibits also peripheral capacitors. Additional capacitors of the surrounding circuit have to be taken into account too. With decreasing diode area, the photocurrent therefore reduces more acutely than the capacitor, which in turn diminishes the achievable voltage difference. Moreover, the influence of area-independent noise terms increases so that the signal-to-noise ratio (SNR) sinks. In consequence, the size of the photodiode determines the dynamic range of a pixel.

Multichannel Photosensors

The spectral sensitivity of a pn-junction depends on the implantation depth. Photodiodes realised at different depths can be used to build colour sensors, which offer additional colour channels for an augmented colour correction. The depth is controlled by the implantation profile. Accordingly, well-substrate diodes and diffusion diodes are available. For the $90nm$ manufacturing process considered here, the junctions of the well-substrate and diffusion diode, resp., lie at a depth of about $1\,\mu$ and $200\,nm$, resp. These diodes are available with any CMOS process; the location of the space charge zone differs, however, with the processes.

In a triple-well process with isolated p-wells another diode can be realised [76, 77], the junction of which lies in about $700\,nm$ for $90\,nm$ technology. These diodes can be arranged laterally as well as vertically (Figure 12.17). In this way more colour channels are obtained per pixel with the same optically

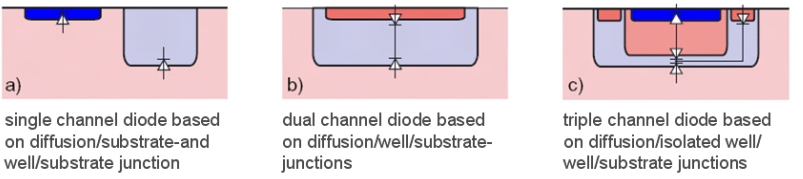


Fig. 12.17 Structures for photodiodes in a CMOS process

active area. Note that with a triple-well structure a fourth diode formed by a n-well-(p+) junction can be realised. Since this diode lies at a similar depth to an p-well-(n+) diode, similar properties can be expected. Hence, it is not useful for the realisation of another colour sensitive channel.
As mentioned before, the prerequisite for the common usage of these diodes is the provision of similar absolute sensitivities of these devices in order to obtain an optimal evaluation of the diverse colour channels. To this end, the photodiodes have to exhibit similar reverse thicknesses, ie. doping concentrations. However, a standard CMOS process does not allow for this. The doping concentration of the diffusions usually exceeds the doping concentration of the substrate and the wells by an order of magnitude. Accordingly, highly qualitative multichannel photodiodes can not be expected with standard processes. For photosensors of high quality, additional process steps with at least 2 more masks are required to allow for additional implantations.

Furthermore, a suitable selection of colour filters is necessary in order to use the potential of colour correction by means of spectral reconstruction as well as to compensate for the imponderabilia of the design. The spectral properties of the photoelectronic devices are essentially determined by the technology. When designing sensors with the help of the $three-domains$ method, the Luther condition must be fulfilled by the colour filter and the sensor [78]. Principally colour filters can be calculated that allow for an exact reconstruction of the required characteristics. The exact copy of the CIE characteristics or of an equivalent linear combination requires, however, quite complicated methods, which do not lend themselves to miniaturisation. Thus for mosaic filters there are no arbitrary characteristics for colour filtering.

By nature multichannel colour sensors make fewer demands of the design of colour filters since the higher parametric dimension of the sensor space can be used for colour correction. This can be taken advantage of when designing filters. For instance, whereas colour filters for conventional photodiodes are wide-band and correspondingly their attenuation is high, those for multichannel photosensors are narrowband and exhibit much less attenuation. In this respect it is to be noted that the problem of the amplification of the sensor noise of strongly correlated sensor channels is not critical at all, since the photonic noise usually dominates the sensor noise.

Sensors with narrowband filters are only a little challenged as to their dynamic range and hence show a strongly reduced SNR. A correlation between the various colour channels is also of advantage for the so-called „demosaicing". With conventional colour sensors the accuracy of the image recording is determined solely by the filter characteristics, whereas with multichannel colour sensors the characteristics are adjustable by means of the correction matrix quite independently from the spectral properties of the colour filters. In the conducted analysis an adequate reconstruction of the CIE standard observer was achieved with eight independent sensor channels. Generally, the combination of diverse colour filters and photodiodes offers a large space of features the user may select from.

Integrated CMOS Pixel Circuits

The charge stored on the photosensitive device must be read off and evaluated. To do this, several methods exist which are based upon CMOS or CCD devices[3]. Analogue CMOS offers to amplify, to transform, and to digitise the photocurrent and the accumulated photocharge, resp. [79]. Note that sensors and the other processing devices can be integrated together, in contrast to CCD technology [80, 81] (Figure 12.18).

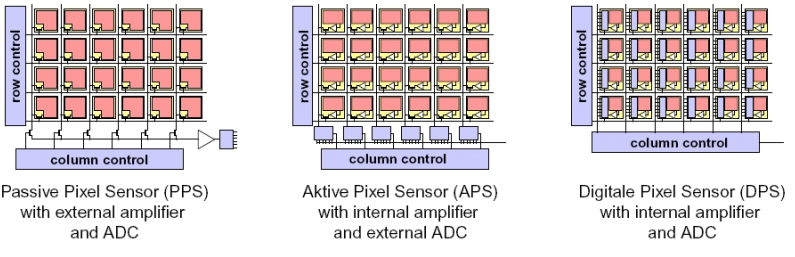


Fig. 12.18 Variants of constructing a CMOS image sensor

Because of the technological shortcomings of the early manufacturing methods, the CMOS technology was not really capable of processing the small signal levels of the photosensors. The problem was caused by the parameter variations of the used devices with regard to threshold voltage, dark current, leakage current, and parasitic capacitors. The resulting noise is called fixed pattern noise (FPN). The bad SNR of CMOS lead to a concentration on the CCD technology, which lasted for about 10 years before CMOS began to dominate photosensors [80, 82, 83].

[3] Charged Coupled Device

Pixel Circuit

Active pixel sensors (APS) built in CMOS technology contain amplifier structures which uncouple the photosensor from the adjacent circuit and thereby provide augmented sensitivity and read off velocity [82, 84]. These amplifier structures are used to design, reset and read-out circuitries. Further elements may be added to account for additional functionality [85].

The standard CMOS process provides three different diode structures which cannot, however, be combined indiscriminately. Note that their quantum efficiency is not easily calculated, since parameters like reflection and absorption coefficients of the layers above the photodiode are unknown and can be measured only when the circuit is ready to be tested. Equally unknown are the doping and the local impurities. Therefore we estimate - on the basis of our experience with preceding CMOS technologies - that the quantum efficiency does not exceed 0.5 (Figure 12.19).

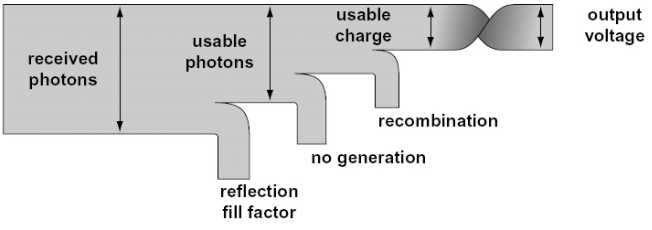


Fig. 12.19 Model to estimate the quantum efficiency

A simple pixel circuit is comprised of a read off circuitry and a reset by a MOS transistor which may use NMOS or PMOS transistors. The largest signal range of the pixel which can be used is noted in Table 12.1.

It should be noted that the common circuit variant with NMOS reset and NMOS read-out offers the lowest useable voltage range whereas the circuits with at least one PMOS transistor experience only a loss of the threshold voltage. To compensate for the voltage loss, with the help of a charge pump,

Reset transistor	voltage across diode	read-out transistor	voltage at input	useable voltage range
NMOS	$0 \ldots V_{DD} - V_{t0n}$	NMOS	$V_{t0n} \ldots V_{DD}$	$V_{DD} - 2V_{t0n}$
NMOS	$0 \ldots V_{DD} - V_{t0n}$	PMOS	$0 \ldots V_{DD} - V_{t0p}$	$V_{DD} - max(V_{t0n}, V_{t0p})$
PMOS	$0 \ldots V_{DD}$	NMOS	$V_{t0n} \ldots V_{DD}$	$V_{DD} - V_{t0n}$
PMOS	$0 \ldots V_{DD}$	PMOS	$0 \ldots V_{DD} - V_{t0p}$	$V_{DD} - V_{t0p}$

Table 12.1 Dynamic range of some circuits

the gate voltage can be increased. Also to be noted are the breakdown limits of modern CMOS processes, which feature very thin gate oxide at a low operating voltage.

The $90\,nm$ technology provides two transistor types which are useful for pixel circuits: digital transistors with reduced leakage current for operating voltages of $1\,V$ and analogue transistors for input/output circuitry with higher voltages of $2.5\,V$. At threshold voltages of about $250\,mV$ for digital and $470\,mV$ for analogue transistors the maximally useable output voltage ranges amount to $750\,mV$ and $2.03\,V$, resp. provided circuit variants with at least one PMOS transistor for reset or read-out are used.

Because of the property of PMOS transistors to transmit a voltage of $0\,V$ across the photodiode correctly, the non-linearity of the discharging process of the photodiode can be analysed in great detail. Accordingly, the leakage currents of the PMOS transistor in $90\,nm$ technology - the gate induced drain leakage (GIDL) and the drain induced drain leakage (DIDL) - significantly influence the pixel characteristics. Both leakage currents are caused by the small channel lengths of the transistors and their respective doping profiles.

The GDIL current is a tunnel current which flows through the gate oxide when a voltage difference between gate and drain is applied. On the other hand, the DIDL current is comprised of charge carriers which tunnel a few nanometers through the potential barrier between bulk and drain [86]. The leakage currents affect the reset transistor and all storage nodes, which are needed, for example for an electronic shutter.

A replacement of the reset transistors by diodes is possible for single or dual-channel photodiodes, when implemented directly in the photoactive area and enables an augmented area usage. An integration of the reset transistor in the photosensor is also possible, although leakage current is unavoidable then.

The design of pixel structures with an electronic shutter should be undertaken only when the leakage currents have been measured. Therefore, simple pixel circuits were selected for test circuitry. Because of the low operating voltages of the selected technology the preprocessing of the analogue signals proved to be difficult. Therefore, a rail-to-rail amplifier was designed and implemented to support the analysis and measurement.

Reset Circuit

In order for an integrating pixel to function properly, the storage capacitor must be reset at the start of the integration phase to an initial value. This is the task of the reset circuitry. The common realisation assumes a NMOS transistor (Figure 12.20a). Its disadvantage consists mainly of the kTC noise, as well as in the diminishment of the output voltage level by the threshold voltage. The variation of the threshold voltages of the reset transistors with the position of

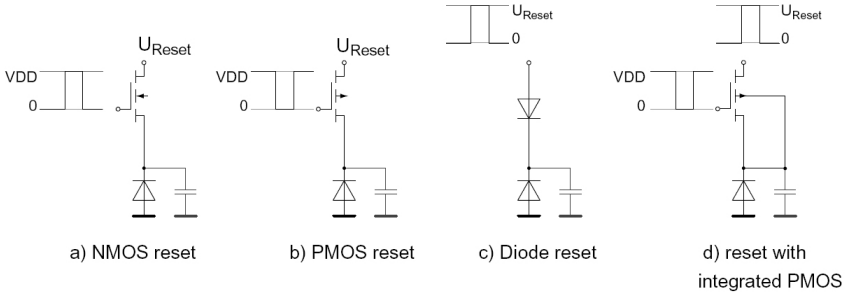


Fig. 12.20 Circuits for different reset variants

the pixels contributes also to FPN. Therefore, an area-saving implementation by means of reset diodes was undertaken, giving similar performance and leakage currents (Figure 12.20c).

The reset voltage U_{Reset} is applied to the diode in the flow direction. The storage capacitor is then charged to the reset voltage minus the flow voltage U_F. After finishing the reset procedure, U_{Reset} takes on the value of a rest potential such that the diode is blocked. In the example of Figure 12.20 the rest potential equals the ground potential. The low reverse current adds to the dark current in this case. To reduce cross-talk from the clock signal, the reverse capacitor of the reset diode must be kept as small as possible.

Compared with a NMOS transistor as a reset switch, a PMOS transistor is not affected by the reduction of the effective reset signal level by the threshold voltage, for PMOS transistors operate in strong inversion when the reset is activated (Figure 12.20b). On the other hand, because of the separated n-well, a PMOS transistor needs much more area and thus reduces the photosensitive area. A new concept is therefore developed in which the reset transistor is directly integrated with the photodiode, for instance in the well of a well-substrate photodiode according to Figure 12.20d. Of disadvantage is the need for additional control signals, which will be generated outside the pixel array, however. The reset transistor can be embedded here for n-wells as well as isolated p-wells, the latter being available with a triple-well process. Figure 12.21 proposes a corresponding circuit including the external circuitry for signal generation.

Read-out Circuit

The simplest read-out circuit consists of an amplifier and a switch. The amplifier accomplishes the uncoupling and the switch selects the pixel located at the bus line. Note that with a source follower, the useable output voltage range is diminished by the threshold voltage.

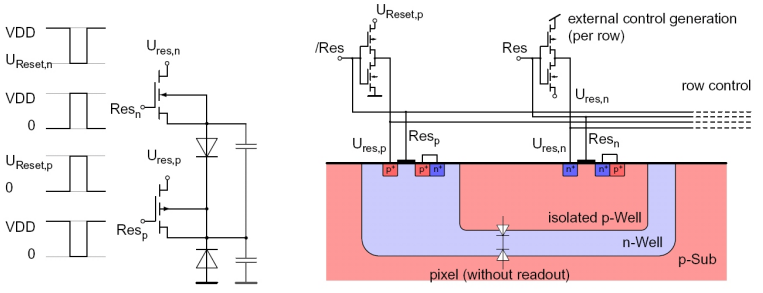


Fig. 12.21 Embedded reset transistor, with substrate cross section

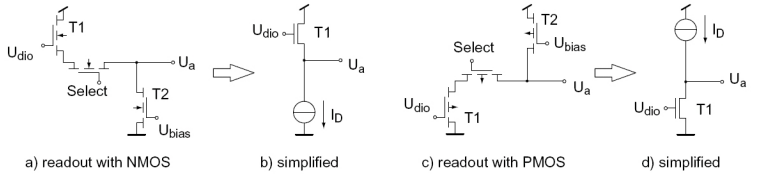


Fig. 12.22 Read-out circuit with source follower

Let us consider a NMOS output stage according to Figure 12.22a. The select transistor T_2 can be neglected in the subsequent calculation, for it is operated in saturation (Figure 12.18b). Assuming $U_a > U_{bias} - V_{T0}$, T_2 can be regarded as an ideal current source.

For $T1$ holds:

$$I_D = \beta \cdot (U_{GS} - V_{T0})^2 = \beta \cdot (U_{dio} - U_a - V_{T0})^2 \ , \tag{12.26}$$

and, respectively:

$$U_a = U_{dio} - \left(V_{T0} + \sqrt{\frac{I_D}{\beta}} \right) \ . \tag{12.27}$$

The output voltage of the source follower, thus, equals the diode voltage minus the threshold voltage and a saturation voltage, which depends on the cross current.

The operating scheme of an image sensor with the above read-out procedure is sometimes designated as a rolling shutter, since the image is read row by row. The integration and the read-out action occurs with a time offset (Figure 12.23). Integration time T_{int} and read-out time T_Z per row determine the image repetition rate $1/T_{image}$, with $T_{image} = T_{int} + T_Z$. The maximal image repetition rate results from the minimal time necessary to read off the entire sensor, neglecting the reset time, and amounts to $n \cdot T_Z$ for n rows. Hence, the maximally available integration time is given by $(n-1) \cdot T_Z$.

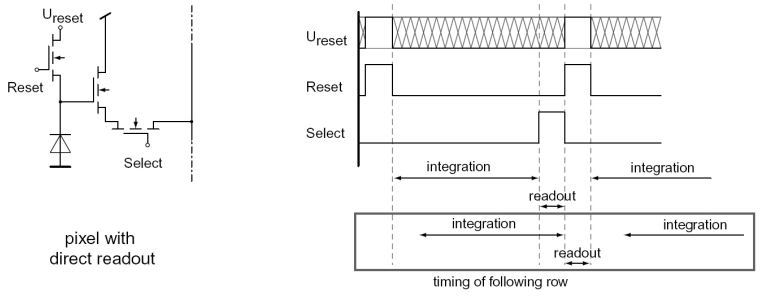


Fig. 12.23 Pixel and read-out circuits, with clock scheme for a pixel with PMOS transistors

It is a disadvantage that the start of the integration phase varies with each row. Higher requirements as to time resolution and uncoupling of integration and read-out phases demand a so-called shutter.

Read-out Circuit with Electronic Shutter

A read-out circuit with an electronic shutter contains another storage capacitor (Figure 12.24). It is realised by an additional device or the gate capacitor of the read-out stage. This capacitor stores the recorded value till the start of the read-out phase and this way enables a concurrent integration time for all pixels. Note that integration and the read-out phase are running independently of each other. Since integration as well as storage capacitor can only be reset simultaneously (Figure 12.24), both phases take place successively. The image repetition rate then reads: $1/T_{image} = 1/(T_{int} + n \cdot T_Z)$.

If the storage capacitor is provided a separate reset, integration and the read-out phase can happen simultaneously. In this case the image repetition rate is determined solely by the integration time ($n \cdot T_Z \leq T_{int}$).

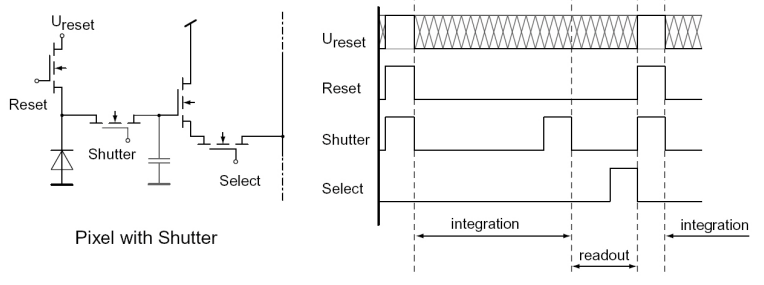


Fig. 12.24 Pixel circuit and clock scheme for a pixel with simple shutter

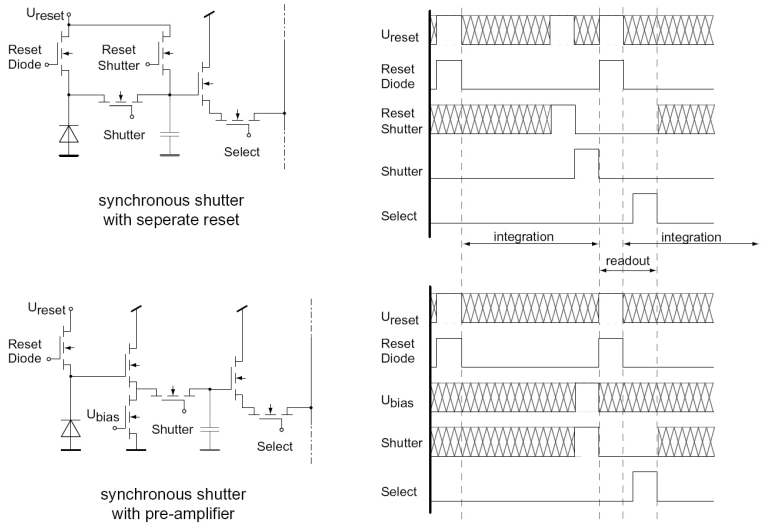


Fig. 12.25 Pixel circuit and clock scheme for NMOS Pixel with synchronous shutter

Another realisation meeting the same functional requirements arises from the usage of an intermediate amplifier (Figure 12.25). In this case the shutter samples the output voltage of the amplifier and does not need to be reset.

Design of Multichannel Sensors in Sub-100 nm CMOS Technologies

For the realisation of single-channel photodiodes, well, diffusion or pinned photodiodes are available (the latter being a parallel arrangement of the first two diodes). Additionally, the doping profile can be influenced with the so-called „moat". In that case the pn-junction between n-well and substrate is extended by a weakly p-doped substrate region from which a larger charge zone space results with a larger number of photogenerated charge carriers. For the reset circuit, variants with a single PMOS transistor, a diode and an embedded PMOS transistor were designed.

Generally, the read-out circuits consist of a PMOS source follower with a PMOS switch (Figure 12.26 to Figure 12.28).

The 2-channel photocells possess separate diffusion-well-substrate diodes, each with their own reset and read-out structures. The designs include different forms of diffusion and moat, in order to analyse the spectral sensitivities and the dependencies of the dark current on the diode type. The reset circuits are designed as PMOS transistors or diode circuits. Read-out happens with the help of a PMOS source follower and a PMOS switch (Figure 12.29).

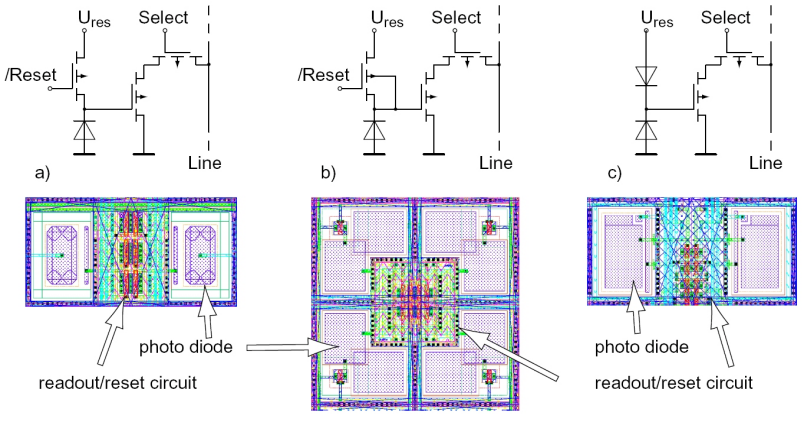


Fig. 12.26 Circuits (top) and layout detail (bottom) of the single-channel pixel cells. Layout details show 2-pixel blocks (a,c) and 4-pixel cell (b), each with combined read-out circuitry.

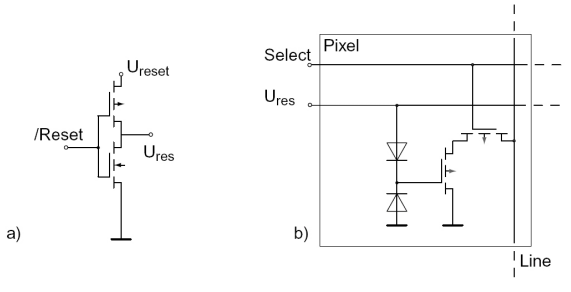


Fig. 12.27 a) Circuit generating the reset signals for the pixel circuit with diode reset, and b) pixel structure with diode reset

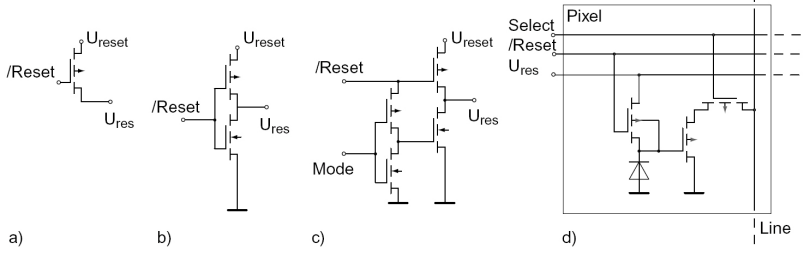


Fig. 12.28 Additional circuits for reset generation in case of inactive reset: a) floating, b) grounded, c) switchable variant and d) pixel circuit with embedded reset transistor

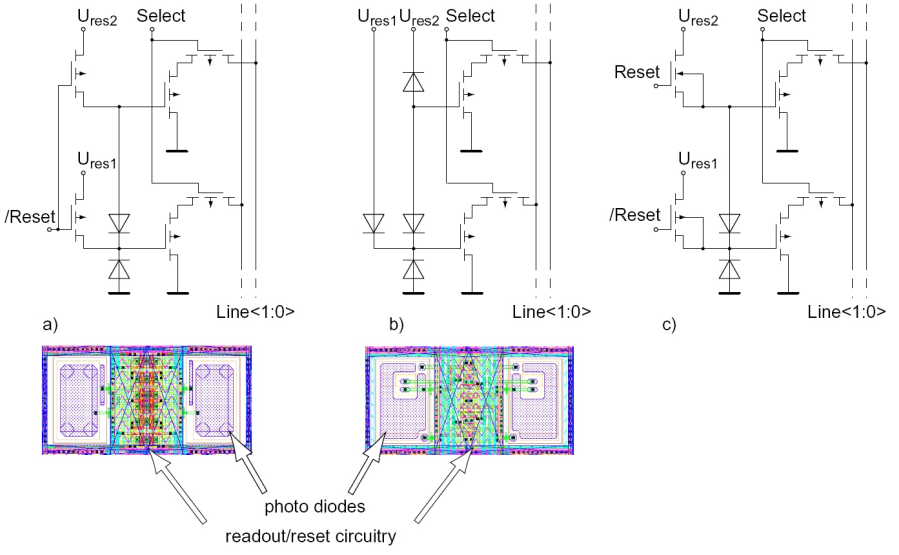


Fig. 12.29 Circuits (top) and layout details (bottom) of 2-channel pixel circuits with (a) conventional transistor reset, (b) diode reset, and (c) reset with two embedded transistors. Layout details show 2 pixels with combined readout circuits.

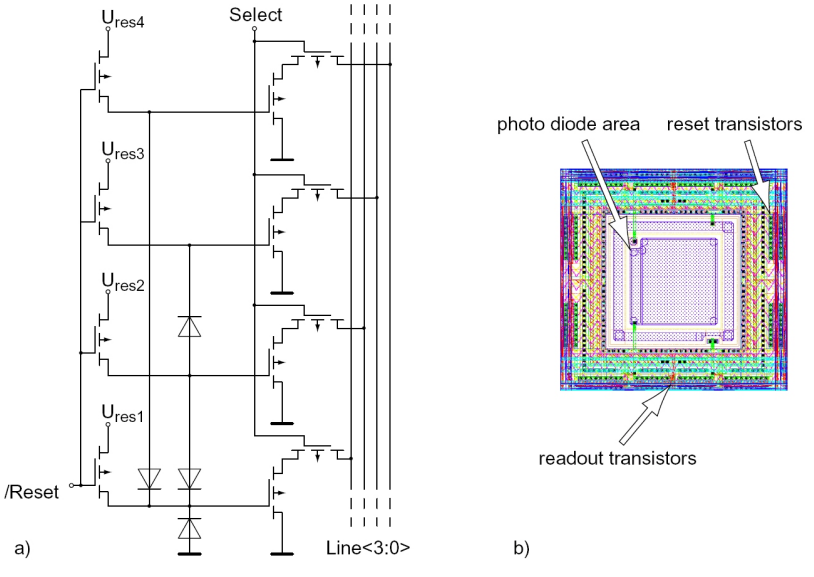


Fig. 12.30 a) Circuit and b) layout detail of a 3-channel pixel with 4 photodiodes

The 3-channel pixel cell has a triple pn-junction composed of a diffusion-p-well-n-well-substrate structure and is available only with a triple-well process. The reset is implemented with PMOS transistors only (Figure 12.29).

The pixel cells were designed for LLD and ANA transistors each[4] and in various forms. Corresponding layouts are shown in Figs. 12.26, 12.29 and 12.30.

Figure 12.31 shows two test chips in 65 nm and 90 nm CMOS technologies with various combinations of the pixel structures described [87].

The images produced by these chips are shown in Figure 12.32a.

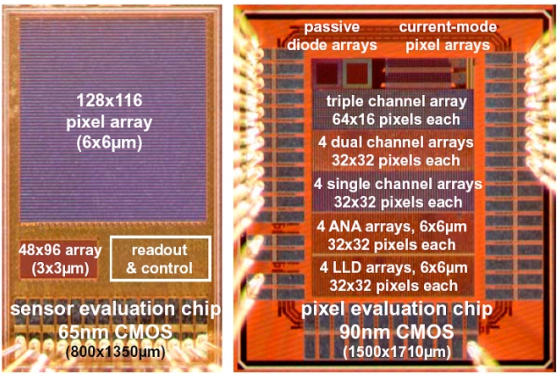
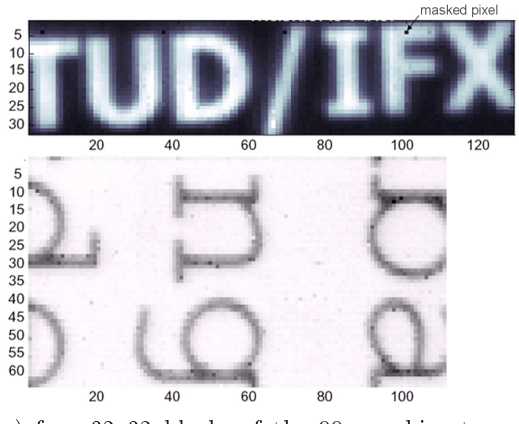
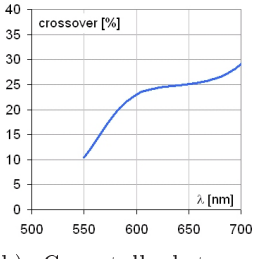


Fig. 12.31 Test chips in 65 nm (left) and 90 nm (right) CMOS technologies



(a) four 32x32 blocks of the 90 nm chip strung together (top), detail of the 128 × 128 array of the 65 nm chip (bottom)

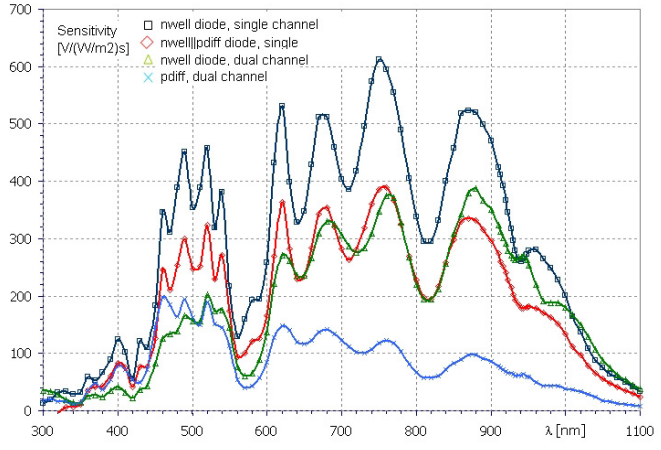


(b) Cross-talk between neighbouring pixels

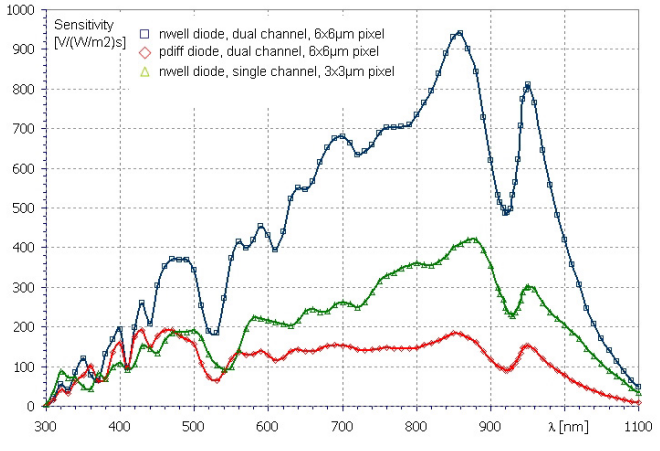
Fig. 12.32 Measurements taken with a camera prototype and the chips from Figure 12.31

[4] LLD - Low Leakage Drain, transistor type with thicker gate oxide and reduced leakage currents; ANA - transistor for analogue applications and IO circuitry

In order to evaluate the cross-talk between adjacent pixels, we masked selected pixels with the help of several metal layers. As Figure 12.32b shows, the degree of cross-talk very much depends on the wavelength. The results of the measurements of the spectral response of the sensitivity of the photodiodes in the diverse configurations of the test chips are documented in Figure 12.33 and Table 12.2.



(a) 90 nm test chip



(b) 65 nm test chip

Fig. 12.33 Spectral sensitivities of the photodiodes

configuration		n-well, moat [3]	n-well/ p-diff, moat	n-well	n-well/ p-diff	n-well, moat	n-well	Ref. [88]
technology		$90\,nm$	$90\,nm$	$90\,nm$	$90\,nm$	$65\,nm$	$65\,nm$	$180\,nm$
pixel dimensions	μm^2	6×6	6×6	6×6	6×6	6×6	3×3	3.5×3.5
integrator capacitor	C/fF	8.2	13.2	12	16.2	11	4.3	4.5
Dark current sensitivity	$R_{dark}/-mV/s$	23	37	210	71	50	7700	6.29
saturation value	N_{sat}/e^-	61 200	98 900	89 900	121 000	82 400	48 400	22 500
reset noise (kT/C) [1]	N_{rn}/e^-	36	46	44	51	42	26	20
dark current noise	N_{dn}/e^-	23	61	315	143	69	2067	4
basic noise [2]	N_n/e^-	43	76	320	152	81	2067	21
SNR	dB	61.2	59.3	48.0	55.9	60.2	27.4	61

[1] we assume $1 \cdot \sqrt{kT/C}$ for reset noise, Ref. [88]: $0.75 \cdot \sqrt{kT/C}$

[2] based upon measurement of dark current and reset noise

[3] lesser doped area between the active regions of pixel cells

Table 12.2 Comparison of selected parameters of various photodiode configurations

Implementation of a Pulsing Pixel Cell

For applications in pulsed neural nets we need a pixel cell which encodes the photocurrent as pulses. We choose a linear transfer characteristics and an area-efficient implementation Figure 12.34.

The main components are the integrator, the threshold switch, and the reset circuitry. The photocurrent is integrated at the diffusion capacitor of the photodiode D_1. Note that the integration is effected as discharge action because of the direction of the photocurrent. If $\bar{U}_{reset} = Low$ holds at reset, D_1 will be charged via D_2. At $\bar{U}_{reset} = High$ (integration phase) the anode of D_2 is grounded, whereby D_2 blocks and the photocurrent discharges the diode

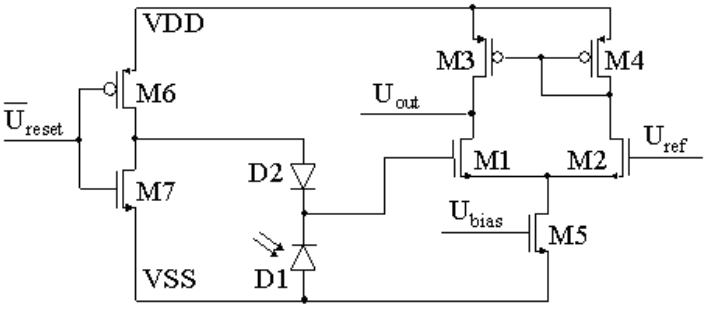


Fig. 12.34 Circuit of the pulsing pixel cell

D_1. As soon as the potential at the gate of M_1 sinks below the preset value U_{ref}, the state of the differential pair M_1, M_2 changes and $\bar{U}_{reset}$ turns into $High$. With the help of a digital delay line (not shown) the minimal duration of the pulse and $\bar{U}_{reset}$ are preset. The total delay till reset is given by τ_{reset}. The period of the output sequence of pulses and pulse frequency f_{out} , resp., then reads:

$$T_{out} = C_{diff,D_1} \cdot \frac{U_{DD} - U_{DSmin,M_6} - U_{D_2} - U_{ref}}{I_{photo}} + \tau_{reset} \ , \qquad (12.28)$$
$$f_{out} = 1/T_{out} \ .$$

The photodiode is charged to U_{DD} at reset. The initial value of the integration amounts to U_{DD} minus the minimal drain-source voltage of M_6 and the flow voltage of the diode D_2. The duration is determined by the photocurrent C_{diff,D_1}, the diffusion capacitance C_{diff,D_1} of the diode, and the reference value U_{ref}. The fixed share of τ_{reset} in the period T_{out} of the pulse, which is caused by some inverter delays and amounts to $3\,ns$, can be neglected when compared with the integration time. The transistor M_5 forces the cross-current of the differential pair and to some extent fixes the operational speed of the switching stage. The reference voltage U_{ref} can be adjusted to the illumination conditions. This pixel cell is designed in $130\,nm$ CMOS technology, and its area amounts to $16,5\,\mu m \times 10,4\,\mu m$. Figure 12.35 shows the layout of the pulsing pixel cell. The digital part of the circuit on the right side comprises an inverter chain for the delaying of the reset signal, circuits for row/column selection, and bus drivers.

Figure 12.36 shows some results measured for the pulsing pixel cell. The upper curve of image (a) displays a single output pulse of the pixel cell and the lower curve exhibits the voltage across the integrating photodiode capacitor. The voltage increases linearly, and about $4\,\mu s$ after reset, the voltage is back to its rest potential. The positive charge curve above results from an

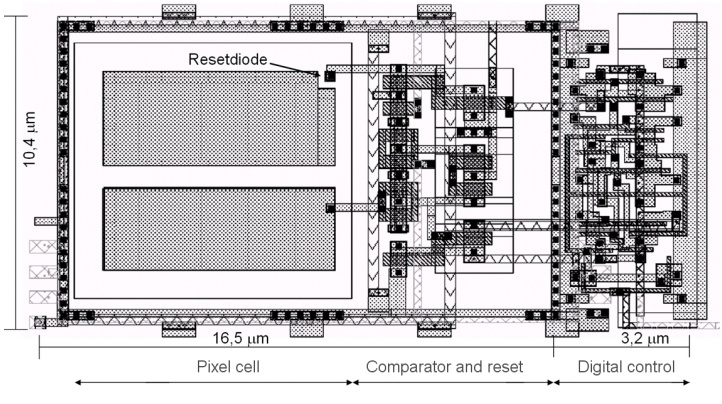
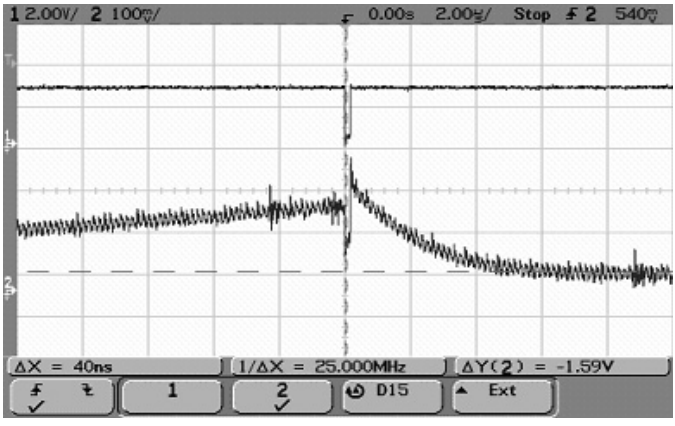
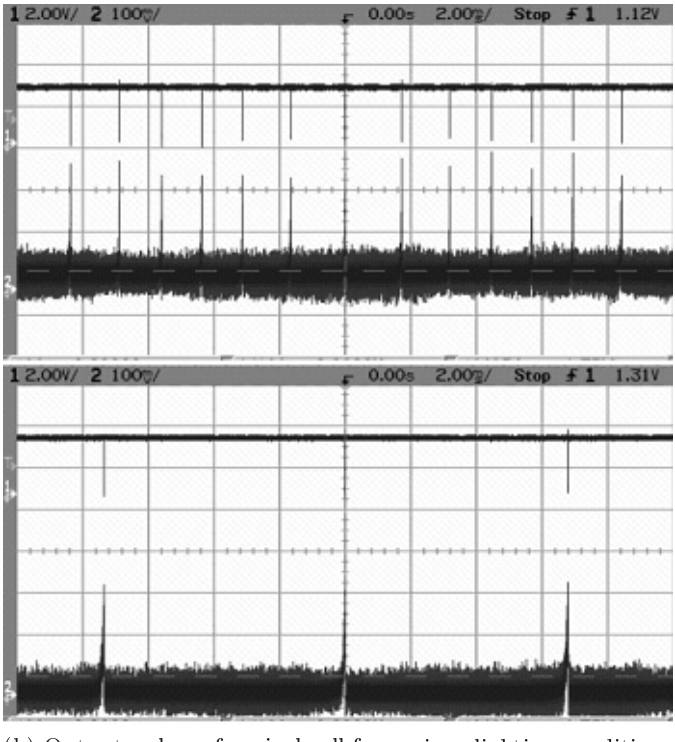


Fig. 12.35 Layout detail of pulsing pixel cell



(a) A single integration and reset step (bottom) and output pulse (top) of a pixel cell



(b) Output pulses of a pixel cell for various lighting conditions

Fig. 12.36 Measured results of the pulsing pixel cell

inverting buffer OTA in the read-out path, and the diffusion capacitance itself is discharged.

With in-room lighting conditions (about $70\,lux$) a pulse frequency of $800\,Hz$ results, with half-light ($10\,lux$) about $140\,Hz$, and with bright daylight ($200-300\,lux$) about $2-4\,kHz$. Hence, the pixel cell behaves as a linear photocurrent-pulse converter, and its power consumption amounts to $110\,nW$, with a current of $50\,nA$ and a voltage of $2.2\,V$.

12.4 DA-Converters and Analogue Image Storage

The integration of the photosensor array and the feature detectors is not possible because of too low a yield. Hence, the images must be digitised, stored and then distributed to the neural net. To implement DA converters in each cell of the neural field would not only take up too much area but also interfere with the analogue signal processing. Instead, the incoming 6 bit word stream of the image controls a bank of current sources. The basic principle of this solution is shown in Figure 12.37.

At the fringe of the neural field a bank of 6×128 current storage cells (SI) is implemented (128 rows with 6 bit resolution each). By means of appropriate current dividers, a reference current provides the binary-weighted current values which are stored and cyclically refreshed in each row of the bank. In each row the image information is applied as a 6 bit word to control the selector switches of the current bank. Consequently, depending on the position of the digitally controlled switches in all rows, 128 sum currents are fed into the neural array at a time and stored again in the current storage cells of the neurons of a particular column of the neural field. Hence, by successive selection of all columns a full image can be stored and processed in the neural field. The image repetition rate is determined by the settling time of the current storage cells of the neurons.

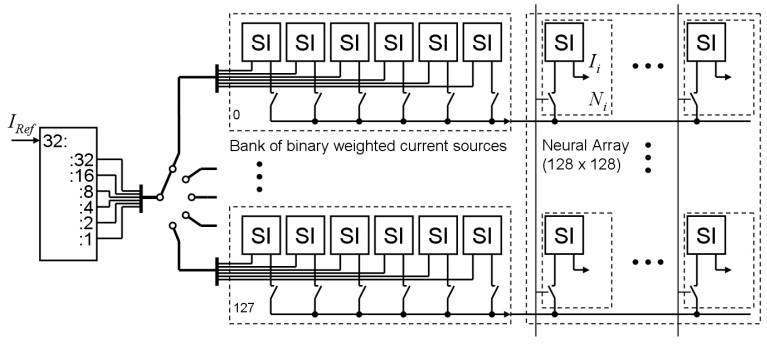
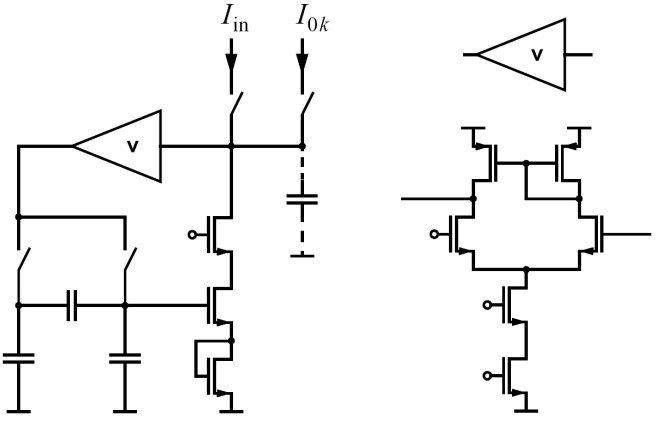


Fig. 12.37 Block diagram of the DA converter for the input and storage of an image



(a) entire circuit with amplifier and storage capacitor

(b) amplifier circuit

Fig. 12.38 Circuit of the current storage cell

The circuit of a current storage cell (Figure 12.38) comprises an amplifier in order to guarantee a short settling time when the current to be stored is small. The maximal value for the settling time is 5 μs. Hence, the 128 column matrix can be refreshed with a repetition rate of about 1000 images per second. The shift from the I_{in} sample phase to the hold phase with output I_{k0} is digitally controlled. The π-like arrangement of the storage capacitors and switches at the amplifier output minimise the interference with the control signal for the switching of the phases.

12.5 Summary

As neurons and synapses are the basic components which appear by the millions in a complex net for vision applications, it is essential to take up as little area for each element as possible. Hence, the exact reproduction of the elements' ideal behaviour as specified by the basic equations 2.1a-c must be abandoned, because too much area is required. Instead, a realisation of the elements is sought that provides equivalent behaviour such that the feature and object recognition, resp., is not compromised.

Based upon a strongly coupled set of workings with regard to modelling, simulation, and design „optimally compromised" circuits for neurons and synapses and pulsing photosensors have been developed.

13

Simulation of Microelectronic Neural Circuits and Systems

Jens-Uwe Schlüßler and Jörg Schreiter

In order to find suitable microelectronic representations of neural systems that provide the necessary functionality at the lowest possible costs, the behaviour of the microelectronic system must be simulated. In the analysis of microelectronic systems of even low complexity, however, it is often the case that the behaviour of the system cannot be described explicitly at all, or only at a huge cost, only approximately, only for special cases, or in a combination of some of these constraints. On the other hand, in many cases at least a numerical solution of the system of equations is possible. We use the word „simulation“ to mean the concept and the solution of the numerical task, some steps of which are done semi-automatically by a software program called „simulator“. The solution is designated as „simulation result“.

Various design variants of a neural net's behaviour can be investigated and compared by using simulation. For example, at first a variant of the mathematical adaptation function of the synapses' dynamics can be searched for and evaluated with respect to the lowest area consumption and the provision of equivalent function. Secondly, simulation can determine acceptable device tolerances, etc., and lastly the entire design is simulated and compared to the specifications (here by means of a differential equation).

When the net under consideration is represented by a moderately-sized circuit, the response of voltages and currents at various nodes of the circuit can be determined with high precision with the help of a simulator for electrical networks. Generally, the electrical network is described by a system of algebro-differential equations, the solution costs of which usually increase about quadratically with the number of state variables and equations, resp., of the system. Hence, it is advisable to look for simplifications which reduce the computational effort but not the required precision.

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 233–246, 2010.
© Springer-Verlag Berlin Heidelberg 2010

13.1 Modelling of Neurons and Synapses

There exists a multitude of state variables in an electric network representing a microelectronic circuit, since internal and external charges must be taken into account separately for each device, whether it is intended or parasitic. Our strategy for simplification reads as follows:

- state variables with small variations are replaced by constants;
- fast transients are neglected and the final value is reached without any delay.

Thereby some differential equations turn into algebraic ones [89]. Another possibility to reduce the costs of simulation consists in reducing the complexity of functions contained in the equations of the system.

The models describing the devices in today's circuits are complex and extensive [90]. Accordingly, the system of equations determining the dynamics of a circuit becomes complicated. Moreover, this system is not always available but rests in the circuit simulator [91].

In order to get simplified models, circuits are analysed with the help of analytically manageable, simple models of electronic devices. These simplified models must nevertheless describe the device's behaviour in the operation mode under consideration with sufficient accuracy.

In the case of low operational speed and MOS transistors operating predominantly in weak inversion, ie. below the threshold voltage, the DC model of Enz, Krummenacher and Vittoz [92] is appropriate [92], which is abbreviated here as EKV DC model. It will be used in the subsequent sections.

Neuron

The transistor circuitry of a neuron is shown in Figure 12.5. The threshold switch is equipped with a positive feedback for fast switching. The feedback is dimensioned such that the switching times are negligible. Thus, the state of the threshold switch can be modelled with a discrete variable.

The output transistors M_{N14}, M_{P11} are operated with a low current in the Ohmic range or are switched off. Therefore the voltage drop across the transistors or their current is neglected. In this approximation, the output voltage takes on only the values of the operating voltage or zero, and there is no need for a state variable of its own.

The transistors M_{N6} and M_{P5} act as switching transistors. Depending on the switching state of the threshold switch, either M_{P5} is on and M_{P4} is operated in the saturation mode with a drain current which is approximately constant, or M_{P5} is off and M_{N6} is on. Hence, it holds: $I_C = I_{\text{in}}$. Then I_C is an algebraic variable only:

$$I_C = \begin{cases} I_{M_{P4}} & x = 1 \\ I_{\text{in}} & x = 0 \ , \end{cases} \tag{13.1}$$

with $x = 1$ designating the switch state of the threshold switch at $U_p \approx U_b$.

For $x = 0$ the input voltage equals the voltage U_C across the capacitance. For $x = 1$ the input current I_{in} rapidly charges the intrinsic capacitances of the switch transistors and the parasitic capacitances at the input to the value U_{in} such that I_{in} becomes zero. In summary:

$$\begin{aligned} x = 0 &\Rightarrow U_{\text{in}} = U_C \\ x = 1 &\Rightarrow I_{\text{in}}(U_{\text{in}}) = 0 \ . \end{aligned} \tag{13.2}$$

The dependency of the input current I_{in} on the input current U_{in} results from the behaviour of the attached circuitry for the synapse.

Synapse

With the help of simplified, analytically manageable transistor models [92] the circuit in Figure 12.13 is analysed. From this the implemented adaptation function follows:

$$\dot{w_{kl}} = -\gamma_0 \cdot (1 - e^{\gamma_1 \cdot w_{kl}}) - \gamma_2 \cdot w_{kl} + \begin{cases} \mu_0 \cdot \tanh \mu_1 (a_k - \theta/2) \ , & x_l = 1 \\ 0 & \text{otherwise} \ . \end{cases} \tag{13.3}$$

The parameters $\gamma_{0,1,2}$ and $\mu_{0,1}$ of this function are derived from the simulation of the transistor implementation of the adaptation circuit.

13.2 Results of Modelling

The suitability of the simplified models for neurons, synapses, and adaptation circuitry is checked by comparing with a complete and true network simulation. For this purpose the time behaviour of a net of 5×5 neurons is considered (Figure 13.1).

Figure 13.2 shows the time response of the pulse events obtained with true and simplified modelling. Because of the approximate implementation of the synapses, in particular the adaptation function, the simulation according to (13.3) differs from the one obtained with the ideal model (11.18). As both responses exhibit synchronisation effects, we proceed to statistical measures to assess the degree of synchronisation.

Let us consider an ordered pair of neurons and their respective pulse sequences and use the standard deviation $\sigma_{\Delta t_f}$ for the time distances between pulses of the pairs as the indicator of synchronisation for the pair. For any subset of the neural net we can then define a total figure of synchronisation for all pairs contained in that subset by determining the distribution of the corresponding standard variations $\sigma_{\Delta t_f}$.

We use test images with known segmentation and measure the distributions after synchronisation has become stable. The pairs lying in the same

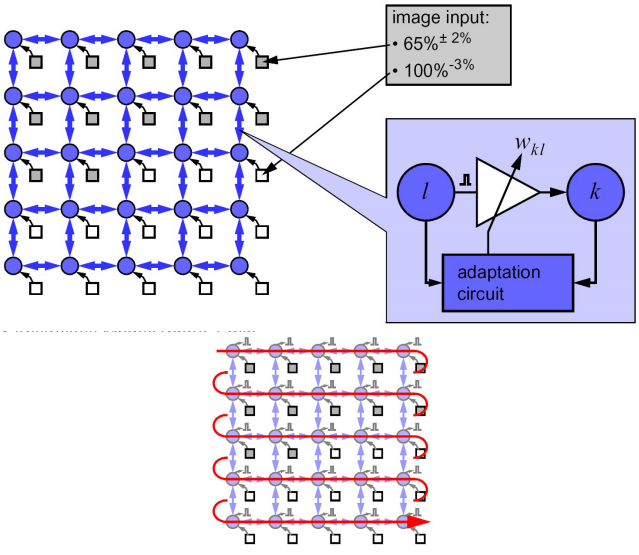
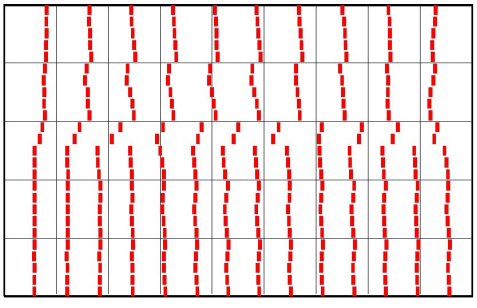
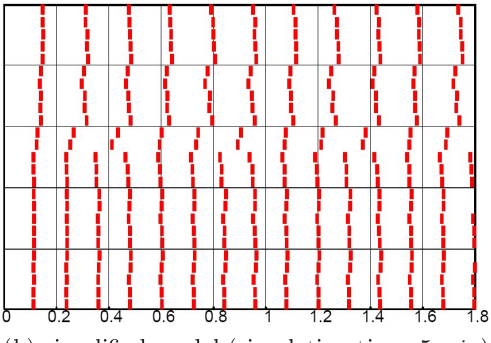


Fig. 13.1 Block diagram and stimulation of the simulated network



(a) Circuit simulation based upon extracted layout data (simulation time: 5 *days*)



(b) simplified model (simulation time: 5 *min*)

Fig. 13.2 Comparison of pulse times for true and simplified modelling

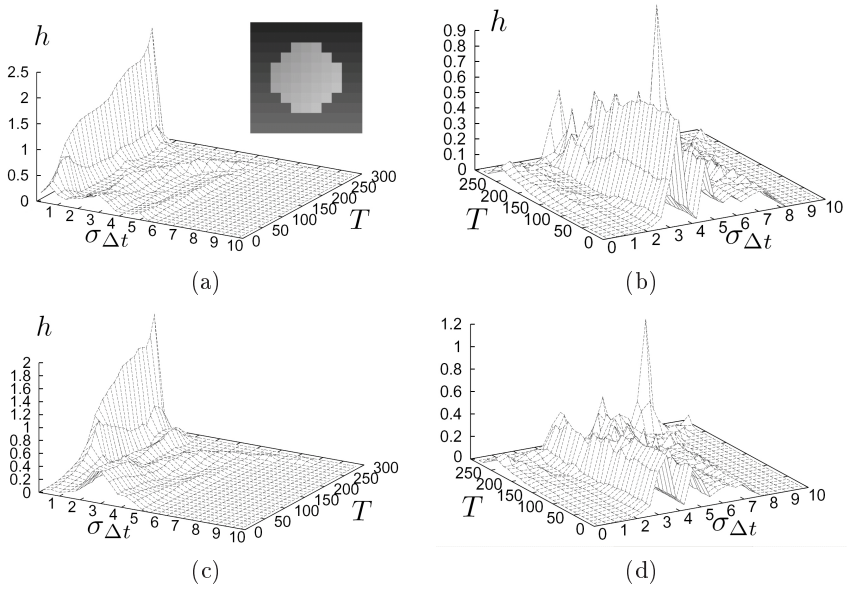


Fig. 13.3 $\sigma_{\Delta t_f}$ histograms of pulse distances. Considered are all possible pairs of neurons with both neurons inside the spot (left column, a, c) or one neuron inside and one outside of the spot (right column, b, d). The ideal behaviour according to (11.18) (top row, a, b) is contrasted with the one of its implementation according to (13.3) (bottom row, c, d).

segment synchronise and, hence, should exhibit very little standard deviation, whereas pairs stemming from different segments should possess a large standard deviation. The distributions resulting from these simulations are shown in Figure 13.3. The synchronisations are considered equal if the distributions coincide or only deviate a little.

13.3 Notes on the Simulation Procedure

Most biological neurons communicate by means of discrete events, namely action potentials. The action potential is generated in the interior of the neuron by time-continuous processes and is modelled by a time continuous system of algebro-differential equations. The triggering of the action potential and pulse firing, resp., results in this model from the respective state variable overstepping the threshold at a particular time. At these discrete times, some parameters or functions may be not continuous, or a description by a different system of equations may hold straight after these times.

In the abstract neuron and synapse models of section 11.3 as well as the concrete models represented in Eqs. (12.1-3), the system of equations changes

also at or after these discrete times. Generally, discrete times may be given as inputs, result from threshold conditions or coincide with the state transitions of some discrete state variables. In the microelectronic realisation, often nets are considered in which the interaction of neurons is effected by discrete signals. In this common situation, the continuous components of different neurons don't interact. In the simulation, ie. the numerical computation of the response of state variables, all these discrete transitions – state transitions of discrete state variables, points of discontinuity of time-variant parameters and overstepping of thresholds – are treated as discrete events. The entire system to be simulated is thus divided into two parts: an analogue system with time continuous state variables and a discrete system which describes the propagation, concatenation, etc. of the discrete events and state variables.

Numerical Integration with Discrete State Transitions

Between discrete events the discrete state variables stay constant. Then the analogue system can be treated as a normal system with continuous parameters. Accordingly, the response of the continuous state variables is determined as usual, ie. by explicit and numerical, resp., solutions of the respective initial value problem of the system of algebro-differential equations.

At each discrete state transition, the system of algebro-differential equations changes, and in the subsequent time interval a new initial value problem is solved. Some state variables stay unchanged at the state transition, as, for instance, the charge of a capacitor in a circuit, which may comprise inter alia switches or disruptively changing current or voltage sources. In the new initial value problem, the initial values of these state variables equal the limit from the left at the right boundary of the past interval. For the continuous state variables, suitable initial values must be found or set.

In summary, a simple starting point for a simulation program consists in beginning from a discrete event and determining the time of the next one in accordance with the Eqs. (2.1a-c). In between the responses of the continuous state variables are computed. To accommodate our concrete modelling of neurons and synapses by Eqs. (12.1-3), only (2.1c) must be replaced by (12.3).

A modified procedure uses the fact that neural nets are composed of many analogue subsystems for which the next events can be computed separately. The closest one is then selected and the response of the continuous state variables of the separate analogue subsystems computed. This procedure has the following disadvantages:

- In large nets with sparse connections, the probability is still high that events happen approximately simultaneously, although the respective neurons are quite distantly located from each other. Accordingly, the size of the numerical step becomes very small, much smaller than the step sizes would become for the separated subsystems.
- The computational effort is high, since for each analogue subsystem the procedure described above must be performed.

Event-Controlled Analytical Solution

If an analytical solution for the analogue subsystems is possible – for instance, with constant synapses and constant input signals or pulses as input – the event times can be determined directly from the equation describing the crossing of the threshold. When an event and pulse, resp., are received before the computed event of the crossing of the threshold, a new system of equations holds and a new analytical solution is obtained. And so on. This case requires the management of a huge event list, including the removal or addition of events.

General Simulation Method for Implicit Solutions

With time-varying synapses or arbitrary input signals, an analytical explicit solution is not possible. Instead, the solution of the initial value problem for the system of algebro-differential equations describing the behaviour of an electronic neural circuit is obtained with the help of the software packet *DASKR* which makes use of implicit multi-step procedures up to the 5th order and variable step sizes [93]. These procedures are suitable for the efficient computation of stiff differential equations, occurring particularly with electronic circuits, since they remain stable even for step sizes much larger than the smallest time constant in the system.

The software function is applied in the single-step mode. At each program call it performs one step of the numerical procedure and computes the end time t for the selected step size as well as approximations (up to the 5th order) for the state variables and their derivatives. In addition, the zeros of other functions composed of the state variables are computed. This way, times for reaching or overstepping the thresholds are obtained. If there is a zero, the end time t of the procedure step, the state variables and their derivatives at this time are recomputed.

Interaction of Several Analog Subsystems

The times of discrete events, triggered by continuous state variables passing the threshold, are determined in the course of the incremental solution of the initial value problem for the continuous state variables. If an event occurs in an analogue subsystem, it may happen that in another interacting subsystem the next step after this event has already been computed. In order not to compute in vain, generally such steps of analogue subsystems are computed alternately. More precisely, the sequence of computing steps reads as follows:

1. Let a system be composed of M analogue subsystems. In each subsystem an initial value problem is solved the latest discretisation step of which is designated by n_m, $m = 1 \ldots M$. In the common reference system we obtain the times $t_m^{(n_m)}$.

2. In the next step, one discretisation step is computed for that analogue subsystem m' the solution of which has progressed least:

$$m' = \arg\min_m t_m^{(n_m)} \ . \tag{13.4}$$

This way, for each analogue subsystem m holds that a discrete event at t_0, say, always occurs either behind the last computed discretisation step at time $t_m^{(n_m)}$ or in the last computed interval $[t_m^{(n_m-1)}, t_m^{(n_m)}]$[1]. Figure 13.4 gives an example.

3. In all subsystems affected by the event the values of the state variables at time t_0 must be computed. Because of $t_0 > t_m^{(n_m-1)}$ it suffices to store the current discretisation step n_m and the last but one $n_m - 1$. In case the event lies in the lastly computed interval, intermediate values must be computed for the state variables. In the other case, further discretisation steps can be computed until an interval appears containing the event. The intermediate values can be determined by interpolation. At the boundaries of the interval under consideration the values for the state variables and their derivatives are available. Thus, a cubical hermitian spline is selected which coincides with the state variables at the interval's boundaries.

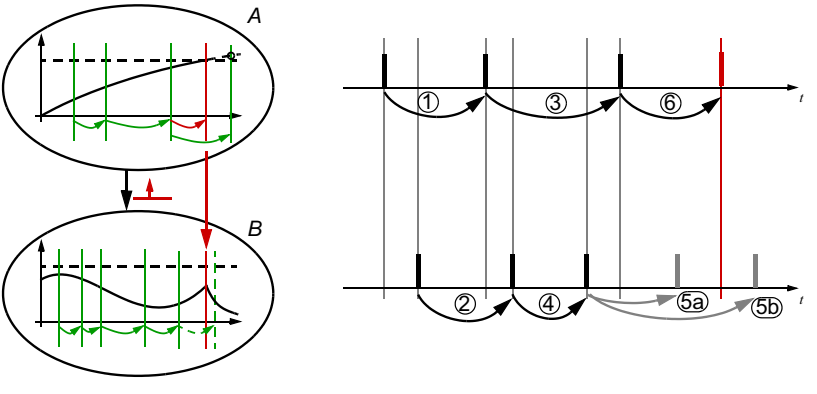


Fig. 13.4 Numerical solution of the initial value problem for a system composed of two analogue subsystems. The next discretisation step of that subsystem is always computed for the elapsed time which lies farthest to the left (steps (1) to 5a and 5b, resp.). With a discrete event occurring in step 6, it lies either behind (preceded by step 5a) or in the last interval (preceded by the calculation of step 5b).

[1] Assume a discrete event is detected in subsystem m' in the course of the calculation of the next discretisation step. Hence this steps leads to the time of the event. $t_0 = t_{m'}^{(n_{m'}+1)}$. Because of (13.4) for all $m \neq m'$ holds: $t_m^{(n_m-1)} \leq t_{m'}^{(n_{m'})} < t_{m'}^{(n_{m'}+1)}$. Then, for each subsystem m the time $t_0 > t_m^{(n_m-1)}$ is an element of the last interval ($t_0 \leq t_m^{(n_m)}$) or $t_0 > t_m^{(n_m)}$.

Since the event has led to a change of the system of algebro-differential equations, a new initial value problem must be solved, starting at time t_0. For state variables not changing their values at this time, the initial values are given by the intermediate values recently computed at time t_0. For state variables occurring at particular intervals, only suitable initial values must be specified.

Validity of Discrete Events

In the case of more than 2 analogue subsystems or feedback by discrete events, additional dependencies develop. Figure 13.5 shows an example of three subsystems A, B and C to illustrate the general case. Let us assume that a discrete event shows up at time t_B in subsystem B which is followed in the course of the numerical procedure by the detection of a discrete event at time $t_A < t_B$ in subsystem A. Therefore, the event at t_B is no longer valid, because at this time a new initial value problem is in existence. Hence, the systems of algebro-differential equations of those subsystems dependent on the events produced by subsystem B (here C) stay unchanged or they change at some other time.

Generally, an event of subsystem m is valid at time t_0 only if no computation of any discretisation step in other dependent subsystems is still pending that could affect subsystem m before t_0. This is definitely the case if in the course of the calculation of (13.4) the subsystem m is selected for the calculation of the next step.

If the subsystem m is selected, the changes in its system of equations can be executed and the first step of a new initial value problem in subsystem m can be computed. Besides, the intermediate values of the state variables of the other subsystems affected by the event described above at t_0 can only now be determined and also new initial value problems can be formulated.

Control Flow for Simulation

The various aspects of mutual interaction of analogue subsystems by discrete events, as compiled in the preceding sections, are combined into a schedule for controlling the calculations of the sequences of the discretisation steps.

All analogue subsystems are listed in ascending order from the times of the last computed discretisation steps. According to (13.4) the subsystem of the first list position with the last computed time $t_{m'}^{(n_{m'})}$ is selected for computation of the next discretisation step, which leads to the time $t_{m'}^{(n_{m'}+1)}$. Hereupon the subsystem m' is filed into the list. All subsystems m with $t_m^{(n_m)} < t_{m'}^{(n_{m'}+1)}$ advance forwards by one list position, whereas m' takes over the released list position[2].

[2] Clearly, in the software implementation,the storage contents containing all the states of the solution procedure are not shifted, but arrays of references are sorted. Or rather, the pointers are changed at the position of insertion and at the beginning of the list.

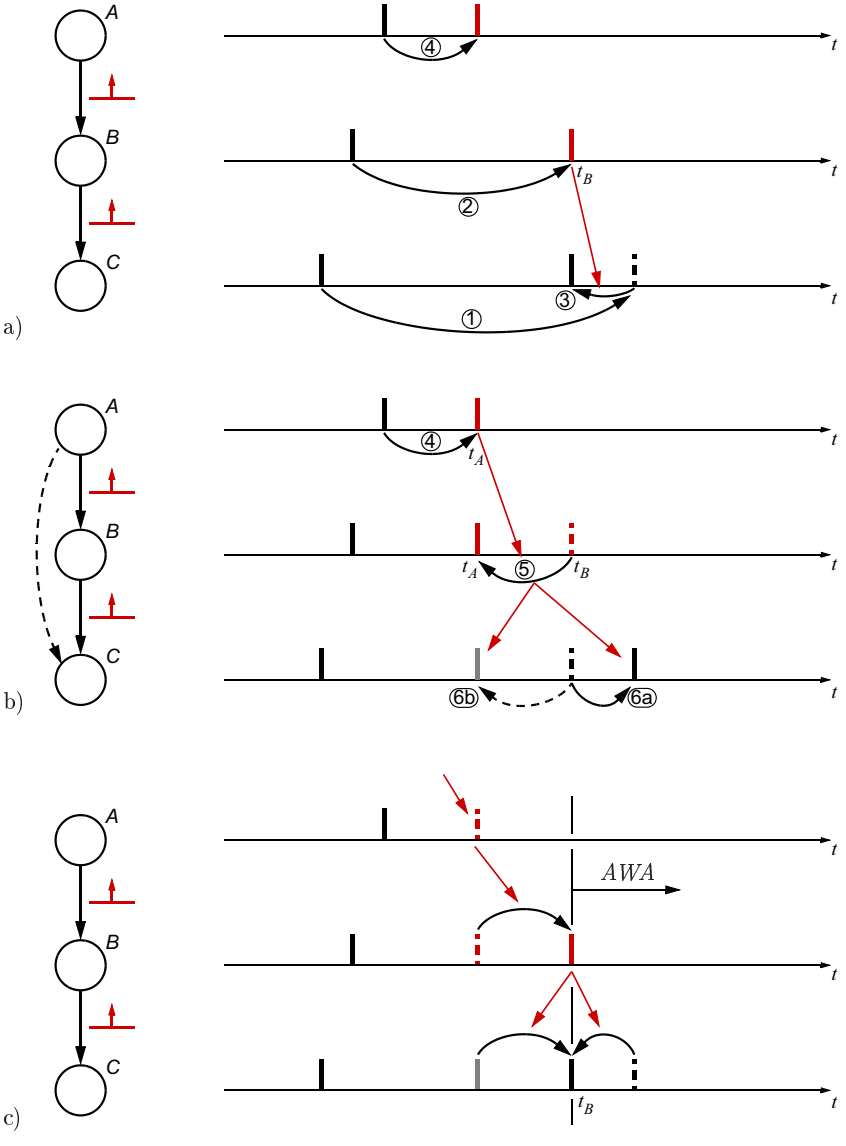


Fig. 13.5
a) A discrete event of subsystem B (2) at $t = t_B$ causes a change of the system of equations for subsystem C (3). b) Caused by an event in subsystem A at $t = t_A < t_B$, the event at t_B becomes invalid. Thereby, either in subsystem C the system of equations is unchanged (6a) or changes at some other time (6b). c) By recursion, a new initial value problem (AWA) can only be begun if no events at earlier times can occur in the course of the numerical procedure.

If a discrete event is detected in the course of calculating the discretisation step for the subsystem m', then m' is marked accordingly and filed in the list as usual. In case a marked subsystem m' is selected for the computation of the next discretisation step, the event which it has been marked for is surely valid. Before a further step can be computed, the intermediate values of the state variables in the subsystems $n \neq m'$ affected by the event must be evaluated. Markers are removed as available, since the affected events do not take place, or at different times. Next, new initial value problems are formulated and all affected subsystems are filed in the list immediately after the subsystem m'. Only then is the next discretisation step of subsystem m' or the first step of a new initial value problem done (in case the event changed the system of equations). Afterwards, the selection of a subsystem according to (13.4) continues as usual.

Complementary Remarks On Control and Application

In certain situations the time of at least the next discrete event can be computed directly:

- for external inputs of discrete events (clock signals, piecewise constant independent sources or recorded sequences of events);
- for analogue subsystems, if the systems of algebro-differential equations are explicitly solvable and the next time of a discrete event is explicitly or numerically computable from the response of the state variables;
- for discrete subsystems, the description of which contains the explicit determination of subsequent events.

With the first and last items, the simulation procedure is complemented by the treatment of external inputs and discrete subsystems. The treatment of particularly simple analogue subsystems can be reduced to the treatment of discrete subsystems depending on the middle item. The control flow does not need any adaptation for the treatment of these cases. The next step of such a subsystem directly computes the next event.

In some subsystems the modification of the system of equations by a discrete event may allow for the immediate release of another event. For instance, the threshold is already passed when selecting initial values, in which case the discrete event produces another discrete event at the same time. This can be taken care of in the schedule simply by marking the subsystem as the activator of an event when the initial values of the next initial value problem are determined. Here particular precautions must be taken care of when new events are continuously released in a discrete feedback loop at the same time.

The schedule is shown in Figure 13.6 and implemented as a software program used for the simulations of the microelectronic networks discussed in chapters 12-15.

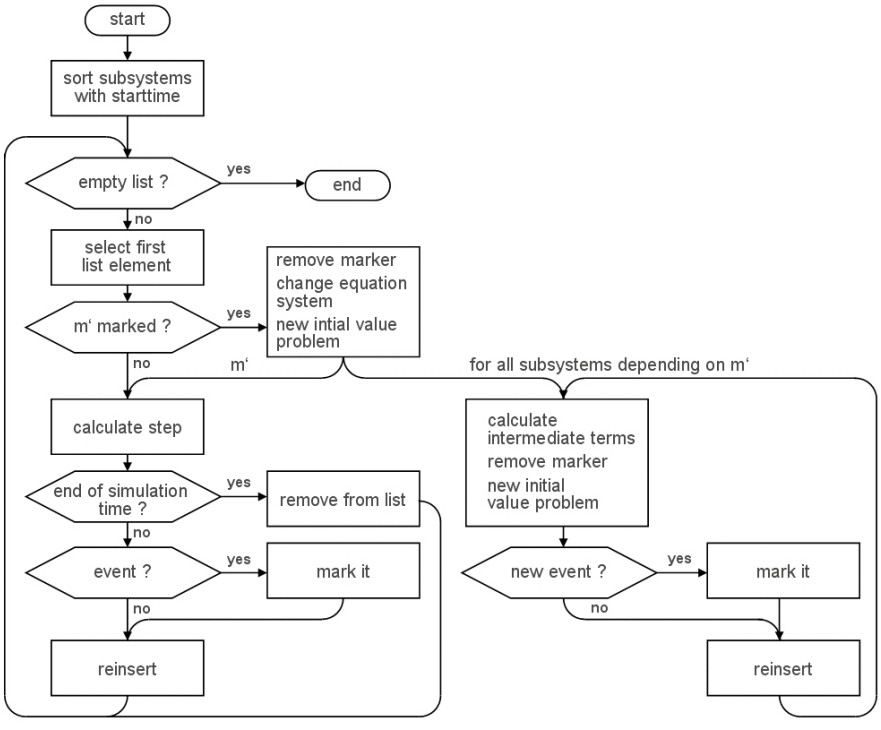


Fig. 13.6 Schedule for the calculation of interacting subsystems

Room for Parallel Program Execution

The software exhibits a single (naturally branching) execution path. Therefore it can not be accelerated by a multi-processor system. Additions are, however, possible which will allow for a parallel execution.

Bulletin Board System

At the highest level of organisation – the control flow for the selection of the subsystem to be treated next – several subsystems instead of a single one can be selected for operation and executed in parallel. For it, a „Bulletin Board System“ is appropriate. It works as follows: a program section selects several subsystems for operation and puts them into an execution list. Multiple instances of another program section select from this list a task, mark the tasks, calculate their results (normally the next discretisation step of the solution of the initial value problem for an analogue subsystem), and replace the tasks with the results and the pointer to the result, respectively. The program section selecting the subsystems must collect these results and note them down in the list of subsystems. With such a procedure it can no longer be assured

that a discrete event discovered in the course of the calculations of a discretisation step lies in or after the lastly computed interval of each of the other subsystems. Thus, several past steps must be stored in order that the intermediate values can be computed if need be. The details of the calculation of a discretisation step only can be abandoned if there is no calculation waiting in the interval subsequent to this step that can overlap with those of any of the other subsystems.

Some of the steps undertaken for subsystems selected in addition to (13.4) are computed for no purpose, because subsequently detected discrete events lie in previously computed intervals. In order to avoid this loss of efficiency, heuristical procedures can be applied for the selection of subsystems.

The advantages of the Bulletin Board System are:

- Multiple processors can execute the program. These processors are relatively cheap and can be arranged in clusters.
- The additional programming effort is low, since the software is already divided into selecting and computing tasks, and open software for the administration of a Bulletin Board System is available.

On the other hand, there exist these disadvantages:

- Some steps are computed for nothing. Heuristical procedures can not completely balance this loss.
- More storage is needed.
- Should the calculation for a subsystem not be performed on one processor, the entire state of the numerical solution procedure may need to be copied to the executing processor unit. Thus, several copies of the procedure are to be stored in the diverse processor units, which causes additional effort in the optimisation of the usage of the bus and memory system.

Parallelisation of the Innermost Loop of the Solution Procedure

At each discretisation step, solving the initial value problem numerically by means of implicit procedures requires the solution of a nonlinear system of equations. For this, Newton's procedure is used, which at each iteration step necessitates the solution of a linear system of equations. For huge analogue subsystems also the linear systems of equations are huge. Therefore, a huge number of computational steps are independent on each other and can be executed in parallel.

The software package `DASKR` [93] does not contain any arrangement for the parallel execution of independent steps. However, the comparable software package `SUNDIALS` [94] exhibits parallelising parts for the innermost loop. On the other hand, this software package lacks the calculation of zeros of additional functions, so it can not be exchanged simply with `DASKR`.

Simultaneous Execution of Independent Simulation Runs

A particularly simple method to accelerate the computation by parallel execution consists of taking advantage of the many independent simulation runs, which differ only in the values of parameters of the system of equations (for instance, Monte-Carlo simulations of electronic circuits). Then so many simulations are available and should be running as processors and computers, resp.

The advantage of this method consists in the very simple realisation, which is carried out in the uppermost hierarchial step of implementation – outside the control flow system – and in the independence of data computed by the diverse processors. For systems sufficiently small to run on a single processor, this method is often the best choice.

Further Potential Augmentations

1. When modelling electronic circuits, usually systems of equations result featuring sparsely occupied Jacobi matrices. The software package `DASKR` unfortunately only offers procedures for densely occupied matrices and matrices with a band structure. Also, `SUNDIALS` has no more to offer.
2. A software package with an accessible function for interpolation in the last computed interval would increase the accuracy of the intermediate values of the state variables.
3. To administrate the well-ordered list of subsystems, the most efficient algorithm for the task „alternating insertion of an arbitrary element and searching for the smallest element“ should be made use of [95].

13.4 Summary

Neural nets have to be simulated at various levels:

1. Level of modelling: neurons and synapses are defined by equations, suitable algorithms selected for the simulation on a computer, and experiments performed which investigate function and characteristics of a neural net. If need be, the modelling is refined in a subsequent step.
2. Level of implementation: neurons, synapses, and nets composed of these are implemented as microelectronic circuits, and per simulation the transfer functions of the respective neurons, synapses, and circuits are obtained. The goal is to reproduce at the lowest cost the characteristics obtained from the first level with the greatest robustness possible.

For both levels, simulation programs have been developed and are available on request.

We have shown that the approximated behaviour of synapses and neurons resulting from the chosen simple and area-efficient implementation provides the synchronisation quality necessary for feature and object recognition, respectively.

14

Architecture and Chip Design of the Feature Recognizer

Jörg Schreiter, Daniel Matolin and Jens-Uwe Schlüßler

The system concept of our vision cube is based on the 3D integration technology the realisation of which is described in chapter 16. The basic concept behind the application of this technology is illustrated in Figure 14.1.

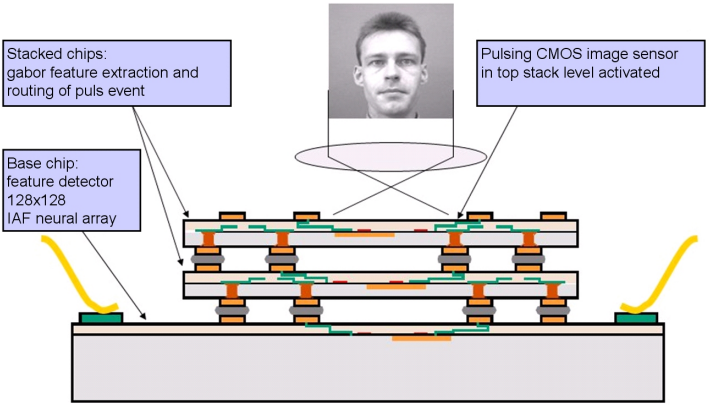


Fig. 14.1 System concept incorporating a 3D integration technology

Several chips are stacked on to a base chip. The stack technology allows the placement of vertical interconnections in high density even in the inner area of the chip. The image information is transmitted optically to the CMOS image sensor integrated in the uppermost layer. Besides the image sensor, the chips stacked at the top contain feature detectors and routers for pulse events. The routers relay the signals from the sensor array or the feature detector to the chip stacked immediately above or below. A feature and object recogniser, resp., is situated in the base chip. The base chip is connected to the chips situated above via vertical interconnects.

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 247–254, 2010.
© Springer-Verlag Berlin Heidelberg 2010

Since the 3D stack technology and the detector and recogniser chips had to be developed simultaneously, the latter two were not optimised for the stack. Instead, their design aimed at

- providing elementary circuitry for feature detection as well as object recognition that can be reused by the classically packaged chips as well as the 3D stack,
- developing a memory-bus-clock architecture meeting the interconnect requirements of the chips mounted on a PCB substrate,
- providing reconfigurability for neural nets used for other applications than feature detection or recognition,
- demonstrating real-time feature detection and recognition with the classically packaged chips.

The next two chapters cover the design of the two classically packaged chips and the results obtained. Note that in chapter 17 there is a description of the architecture of a vision system, which is optimised for 3D stacking and exhibits much more parallelism of the diverse functional units and much lower clock frequency.

We start the discussion with the presentation of the recogniser situated at the bottom of the stack. Strictly speaking, only the inner parts of the feature recogniser are described, as the synchronisation detector and circuitry for object binding are not included in the design (cf. chapter 7-8). In addition, we assume a specific network architecture which evenly covers connections to nearest neighbours within the first circle only. This network can be seen as a superimposing of two feature detectors, one of them rotated by 90 degree. The term „spot detector“ for this network is derived from its all-round characteristic, and its application is segmentation, as described in chapter 5. Thus, except for the feature specific network architecture, a general feature recogniser is on hand.

14.1 Chip Architecture of the Feature Recogniser

Figure 14.2 illustrates the basic architecture of an integrated circuit, which implements the spot detector using a pulse-coupled neural network.

The analogue circuitry for the neurons is arranged 2-dimensionally. An individual pixel from an image sensor is assigned to each neuron. The pixel information is transmitted via an external interface (by vias in the 3D stack).

The block diagram in Figure 14.3 complements the system architecture with the details of the external interfaces. Via the input interface, the pixel data, the initial values of the synapses as well as the parameters of neuron and synapse functions are transferred. Most of these values are digital and converted to analogue format by means of an integrated DA converter (section 12.4). The selection circuitry at the fringe of the neuron matrix deals with each neuron separately. The pulse signals of the neurons can either be conducted via 3D connections to other chips of the stack or to pads.

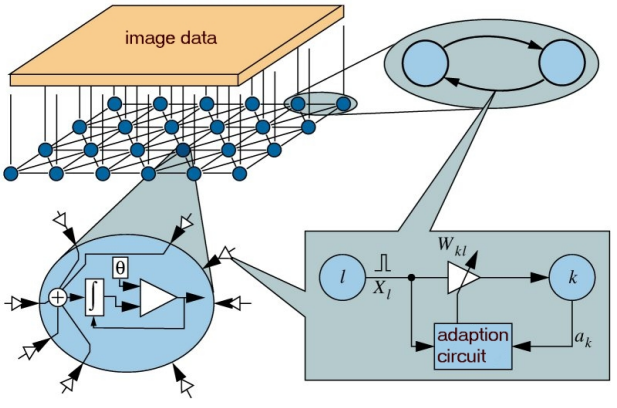


Fig. 14.2 Schematic circuit of the *spot detector* including a neuron circuit (bottom left) as well as a circuit for weight adaption (bottom right).

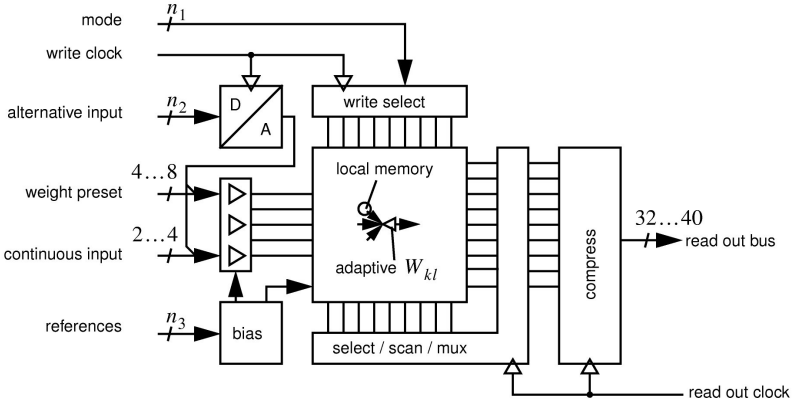


Fig. 14.3 Architecture of the feature recogniser

14.2 Interfaces for Test and Readout

Digital-analogue converters in current mode circuit technology are used for inputting data and pre-adjustment of the synapses (section 12.4). In implementing a complex analogue circuit for the first time, characterisation is necessary and a respective test interface must be integrated (Figure 14.4). This enables the stimulation of individual inputs and the observation of accumulator potentials of separately selectable neuron circuits.

The operational speed of the system is at least 1000 times higher than the biological pulse rates in order to reach a sufficient throughput rate, the limiting factor here being the bandwidth available for the read-out circuits. If

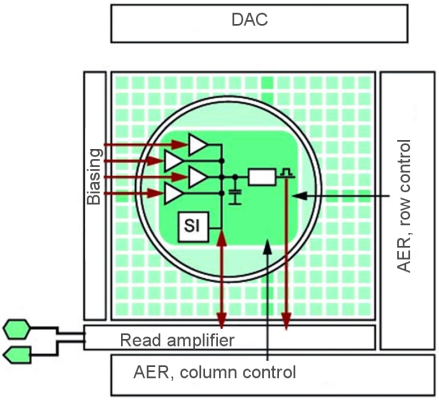


Fig. 14.4 Interface for test as well as for measurement of analogue components

all neurons operate at the highest speed, the bandwidth is about 160 million pulses per sec. Assuming an IC tester with 48 digital channels and a sampling frequency of 100 MHz, a data rate of about 4 GBit/s can be reached. For encoding of time and location of a pulse, 24 bits are then available. Of these, 14 bits are necessary for locating a pulse in a 128×128 field of neurons. Various sampling procedures with subsequent data compression as well as address-event coding methods (AER) have been investigated for the readout of the pulse patterns [96] in order to balance the time resolution and the implementation effort. Accordingly, an address-event protocol has been fixed and a respective RTL description implemented (Figure 14.5).

The coding of pulse events is dependent on the context given by the space-time succession of pulses (Figure 14.6). 2 or 3 prefix bits suffice to select the coding scheme, and the entire neuron matrix is divided into blocks of 4×4 size. A row and a column of one or several events are encoded by 5 bits each. The row code is suppressed if successive events are read out from a column. The coordinates of a single event of a block are encoded by 4 bits. If multiple events are present during the same time segment, a „one hot“ coding method takes care of them (here each bit of the 16 bit code block is attached to a single neuron of the 4×4 neuron block).

The clock frequency of the digital system is set to $120 - 200\,MHz$, from which a time resolution of about $5 - 8.3\,ns$ results for the pulse events. This value is sufficient for pulse rates up to $10\,kHz$. For an optimal clock frequency in a 3D system we refer to the presentation in chapter 17.

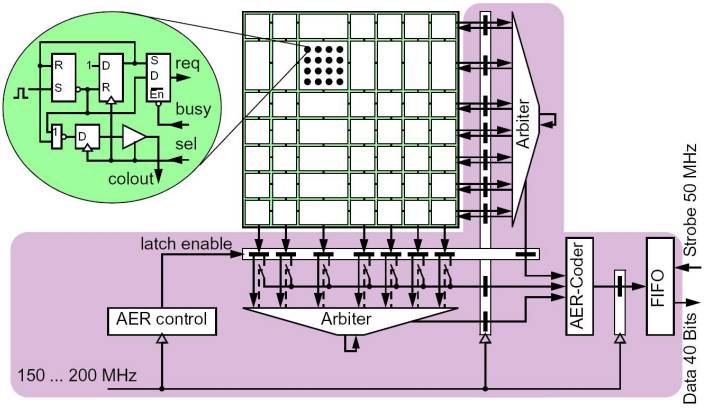


Fig. 14.5 Address event technique (address event representation, AER) for the capture of pulses

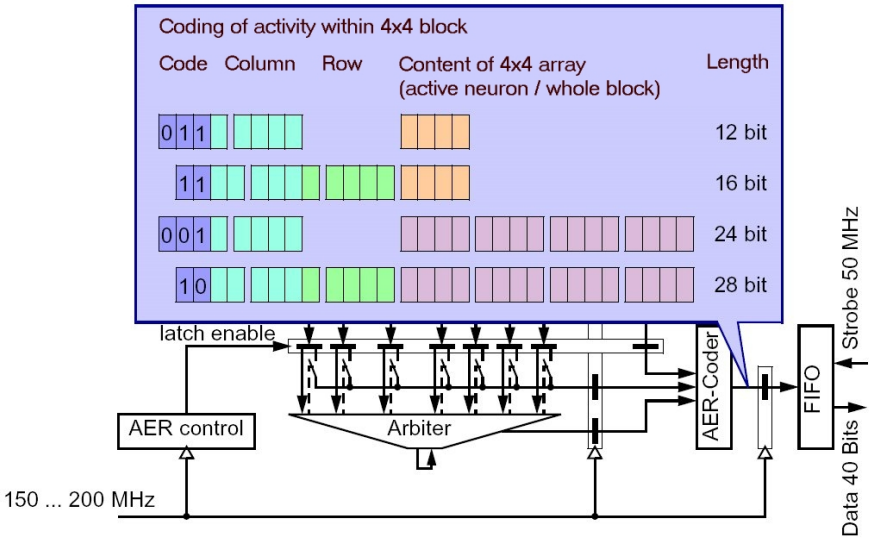


Fig. 14.6 Used code for the AER technique

14.3 Chip Design and Layout

The components described so far (analogue neurons and synapses, test circuitry, digital I/O) were designed and set out as individual full custom macros. Afterwards neurons and synapses were assembled with the help of the layout editor to form the 128·128 matrix, including the interface circuitry. Figure 14.7 displays the chip and layout details, which show the neurons and vias for the 3D stack.

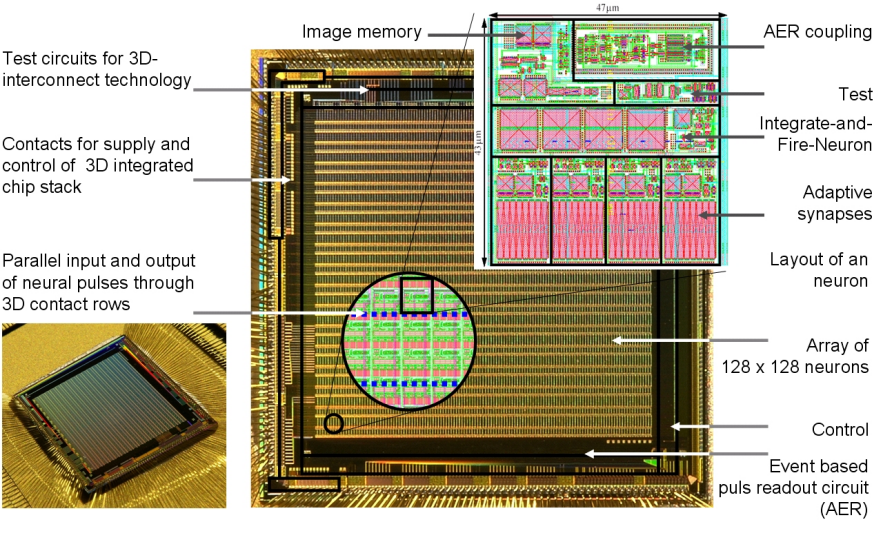


Fig. 14.7 Chip photograph and layout details of the feature recogniser in 130nm CMOS

14.4 Demonstrator and Measurement Results

The chip is designed such that feature recognition can be demonstrated with or without the 3D stack. The configuration of neuron and net parameters as well as the initialisation with image data is performed via an IEEE1394 interface with the help of a computer. The same interface serves to record, store and finally visualise the pulse events on the computer.

As a special illustration a stylised apple is selected (mid left in Figure 14.8). At the top in the middle, as a sign of synchronisation, the pulse wave is running over the apple.

Figure 14.9a displays a short video sequence of a head. The image segment contains 52×52 pixels. Coloured markers indicate the almost simultaneous presence of pulse events. Clearly, the complete video of the propagation of the pulse wave is quite spectacular and proves the segmentation function much

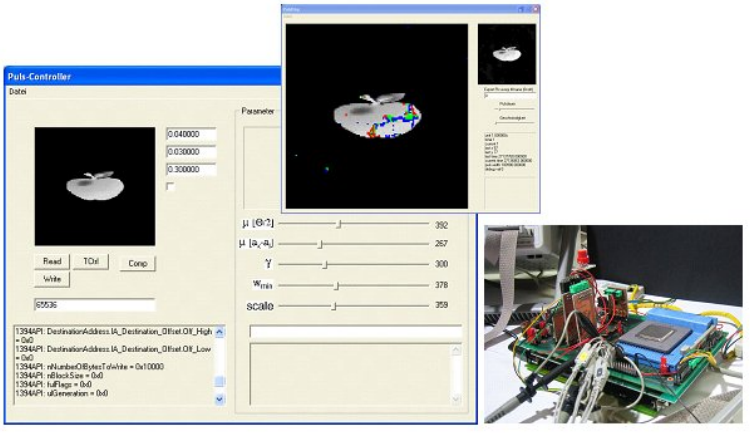
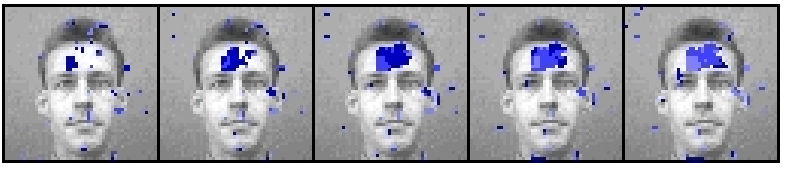
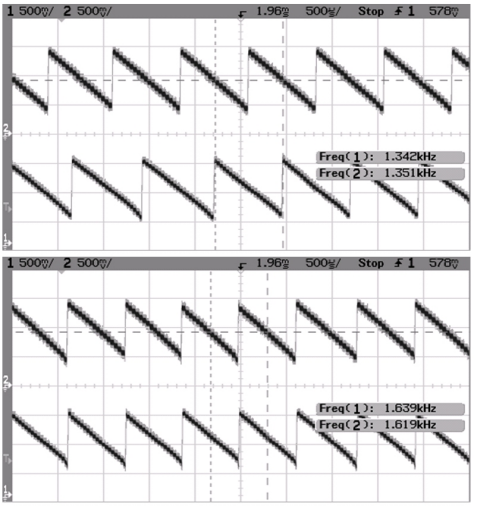


Fig. 14.8 Configuring the neural net (left), visualisation of the pulse sequences (mid), test assembly for the feature recogniser (right)



(a)



(b)

Fig. 14.9 (a) Recorded video section, (b) measurement of the accumulator potentials of two adjacent neurons with adaptation switched off and on, resp.

better. The stimulation of the net was achieved by image data, which were transferred to the chip via the DA converter interface.

At this time we would like to remind the reader of the microelectronic implementation of the adaptation rule (11.19) for synapses, which is different from the ideal synapse behaviour modelled in (11.18). This confirms the robustness of the adaptation rule.

At separate test outputs of the chip, the pulses of freely selectable single neurons are accessible. Figure 14.9b shows the oscillographic recording of the accumulator potentials of two adjacent neurons, the adaptation of which is being switched off and on, respectively. The two neurons stem from the forehead from where a voluminous pulse wave can spread out.

14.5 Summary

A circuit with $128 \cdot 128$ neurons, uniformly connected to adjacent ones via dynamic synapses, has been built of the modules for neurons and synapses. Accordingly, the feature characteristics were isotrop, and the circuit functioned as expected as a spot detector and a segmentiser, respectively: connected pixel regions of about equal brightness caused the attached neurons to synchronise their pulse activities. Since the generic feature recogniser differs from the presented one only by the unisotropic connection - adjustable by programming the connection structure according to the selected feature characteristics (for the details, cf. chapter 7) - with the example of the spot detector we have proven the suitability of our architecture for the generic feature recogniser.

15

Architecture and Chip Design of the Feature Detector

Christian Mayr and Jens-Uwe Schlüßler

This chapter describes the system architecture and chip design of the feature detector. The design aims at a widely flexible configuration of the connection topology to realise arbitrary feature detectors, at a faithful reproduction of the simulated individual or group behaviour of neurons and synapses, and at low power dissipation as well as the integration of image sensor pixel cells in order to directly supply the feature detectors with input signals.

A characteristic of the feature detector chip is the comprehensive use of digital circuits, with the single exception of the image sensors. In contrast, the feature recogniser has been realised entirely with analogue circuits, as described in the previous chapter. In both cases the circuit comes up with the smaller area for the same input-output behaviour of the block which carries out the feature detection and recognition, resp.

15.1 Digital Representation of the Feature Detection

The basic operation of the feature extraction is the pulse difference computation, as realised in Figure 5.8 employing neurons and synapses. If the 128 input pairs are distributed suitably across a small area of an image and if the weighted sum of the pulse rates over all placements is computed (comparable to neuron N_5 in Figure 6.2a), then there is, as we know, a very good correspondence with the Gabor wavelet filters. This means that because of the averaging, all correlations and decorrelations, resp., produced by our excitatory resp. inhibitory dynamic synapses, do not affect the output of neuron N_5. Only the pulse rate appears there. Accordingly, it makes sense to replace the assembly of neurons N_1-N_5 with a difference counter and a weighted accumulator carried out in digital circuitry. The functional equivalence with the simulation results is shown in Figure 15.1.

The pulsed version of the input image is generated via an AHDL behavioural description of the pulsing pixel cell as documented in chapter 12, which is supplied with a current representation of the greyscale input image.

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 255–267, 2010.
© Springer-Verlag Berlin Heidelberg 2010

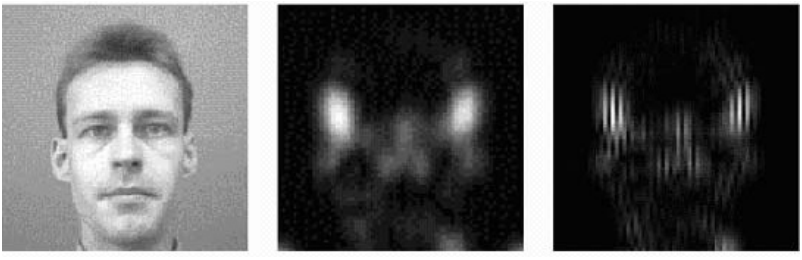


Fig. 15.1 Comparison of original image (left) with continuous-valued (mid) and pulse based Gabor convolution (right)

In contrast to Figure 6.10, the image of Figure 15.1c exhibits less noise. This is due to the fact that the digital representation with counter and accumulator does not incorporate the imperfections of the assembly in Figure 5.8. This imperfection manifests itself as output pulses of neuron N_4 even though both inputs are supplied with the same pulse train. This erroneous pulse rate is only revealed with a low feature contrast.

As can be seen from Figure 15.2, the most significant aspects of the full Gabor answer are visible in the output picture after only 4 output pulses of the brightest pixel have elapsed. The pulse-based image processing chain as presented here in its digital representation thus exhibits analogies to biological neural networks: coexistence of fast, coarse-grained processing and slow, detailed processing [97]. In the following, the term neural processing unit (NPU) will refer to the basic component of the digital realisation of the feature detection, which is composed of the difference counter and the weighted accumulator.

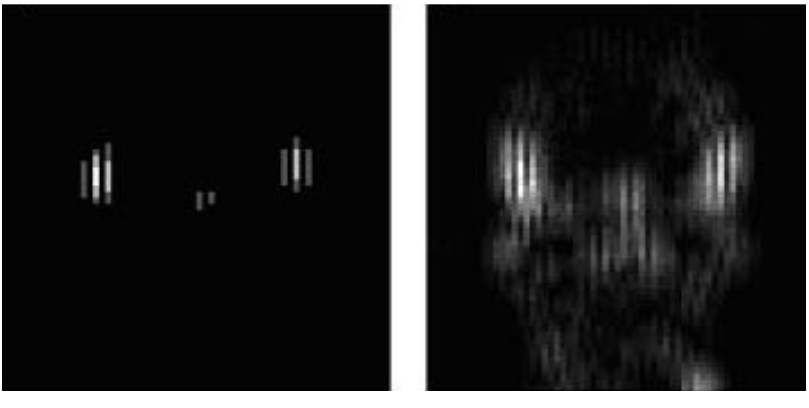


Fig. 15.2 Comparison of the Gabor response after at most four pulses of the pixel cells (left) and complete response (right)

15.2 VLSI Design of a Neural Processor and Router Circuit

System Concept of the Circuit

The whole circuit has been realised as a full custom digital design in 130 nm CMOS technology with automatically generated RAM macros and mixed-signal inserts. Figure 15.3 shows the floorplan of the chip, subdivided into its building blocks. Most of the chip area is occupied by the array of 128×128 neural processing units (NPUs) which will be described in more detail in the following subsection. The NPUs have been synthesized from a high-level behavioural description (*VHDL*).

A place-and-route-tool was used for the placement and wiring of the NPUs. The layout macros of the pixel cells were designed as analogue circuits with manually generated layouts and subsequently positioned in placeholder cells in the NPUs.

In Figure 15.3, arrows denote how a pulse is transmitted via the row and column decoder across the router array to its target NPU. The block

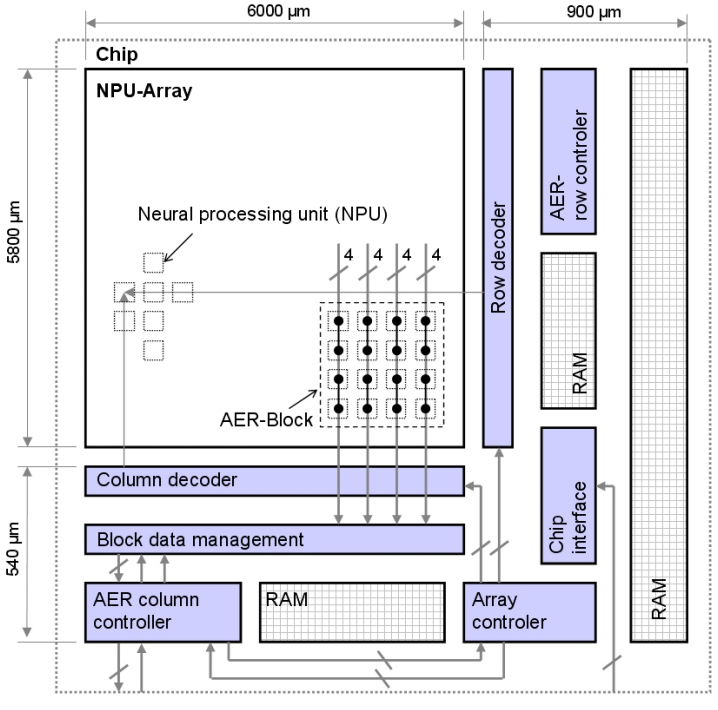


Fig. 15.3 Overview of the floor plan of the feature detector chip with its building blocks

structure of the AER[1] unit and its division into row and column processing on the global chip level is indicated. The configuration of the NPUs and the router memory is done using a JTAG interface contained in the block *Chip interface*. The simulated power consumption for router and periphery for a mean pulse distribution rate of $160 \cdot 10^6$ pulses/s is ca. $2\,W$. The additional power consumption of the NPUs is strongly dependent on their configuration (pixel cells, difference counter or digital neuron/accumulator active); it varies between $100\,mW$ and $1,5\,W$. This power consumption does not compare so badly with the biological original when taking into account the speedup of about 1000 with respect to biological spike frequency.

Figure 15.4 shows an overal view of the chip layout, with additional details. In the detailed view, the photodiodes of the pulsing pixel cells can be seen (black squares), as well as the metal areas of the contacts for the 3D vias (rows in lower third of the detailed insert).

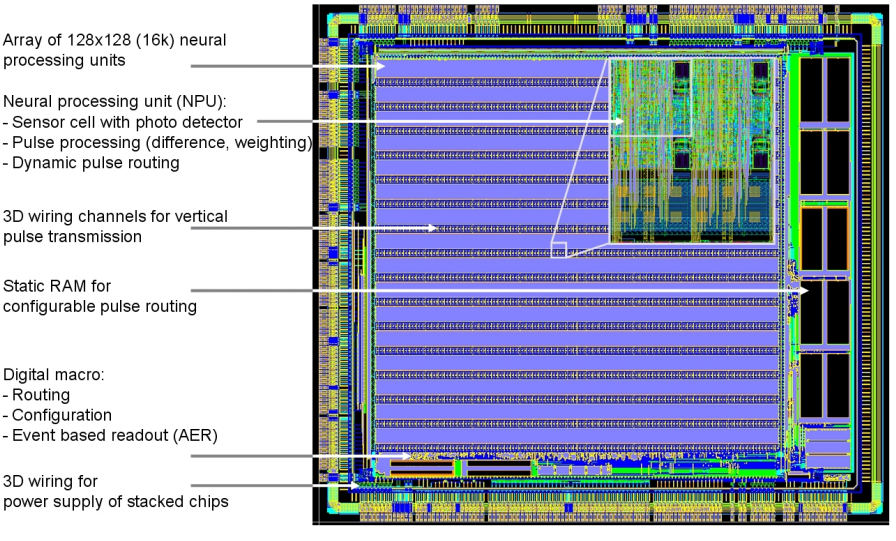


Fig. 15.4 Layout view of the neural processor and router circuit, containing a detailed view of the neural processing unit (NPU). The overall dimensions are $7.7 \times 8.0\,mm^2$.

Neural Processing Unit

The composition of the NPU is shown schematically in Figure 15.5. The NPU contains a circuit for the pulse difference computation („2-before-1“), replicating the function of neurons $N_1 - N_4$ in Figure 5.8, a digital accumulator

[1] Address event representation (AER)

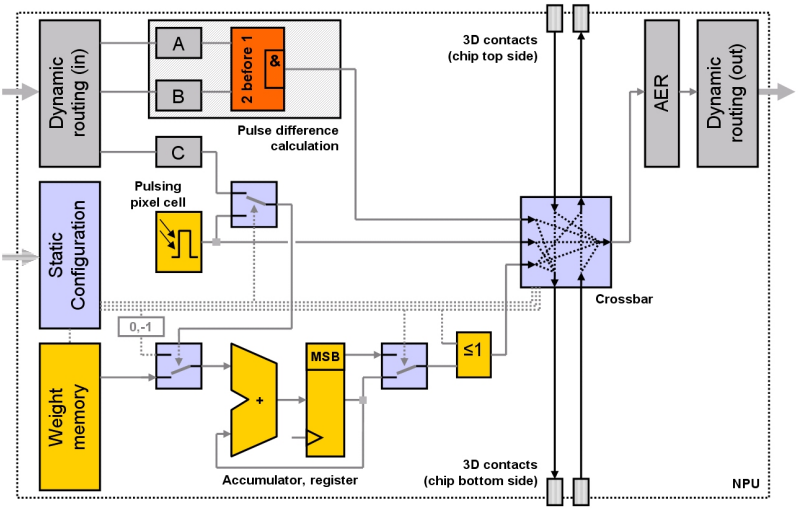


Fig. 15.5 Neural processing unit with circuits for pulse difference computation, a digital neuron, configuration and routing blocks

with configurable input weight (representing neuron N_5 in Figure 6.2a), a pixel cell providing the pulse-rate-coded input signal, and a functional unit (crossbar) for transmission (routing) and coding (AER) of pulse events. Instead of a point-to-point connection between neurons, which is prohibitively expensive for large nets, the router transmits a code word containing the time and place of the pulse generation. The regeneration of the pulses as a rectangular waveform is carried out as needed during the decoding in the target neuron.

The NPUs are distributed across the chip in a regular 128×128 matrix structure to keep the design effort to a minimum and to achieve a neural FPGA-like functionality. Depending on the configuration, single basic blocks can thus be connected together to achieve a complex overall processing function. To keep the realisation of the connection topology freely programmable, a central, fully configurable routing block (dynamic routing, crossbar) has been designed, which receives pulses from the whole NPU-matrix and arbitrarily retransmits them to one or several NPUs. For capturing the pulses across the NPU array, the AER coding unit of section 14.2 has been reused.

The NPU receives pulses from the router and retransmits them, according to the processing tasks, to one of several inputs of the subunits described in the following. Depending on the NPU configuration, the resulting output pulses are then passed on to another chip (in the case of the 3D stack: the chip above or below), or transmitted to the AER unit on the same chip. Further 3D routing is provided through a suitable configuration of the matrix (crossbar) between AER unit, 3D contacts and NPU. For example, pulses can be passed

vertically between the upper and lower 3D contacts in both directions. Via the AER unit, pulses can also be transmitted horizontally, i.e. across the same level in the chip stack, to a 3D contact at a different location.

In Figure 15.9 below, a sample configuration and topology for the NPUs is presented which realises the feature extraction discussed in section 15.1. Different functional units of the NPUs (pixel cell, cicuit for pulse difference computation, summation) are thus used to implement a network for feature extraction.

Pulse Difference Computation

As described in section 15.1, the feature extraction requires a circuit arrangement for pulse difference computation. Figure 15.6 shows such an arrangement based on digital circuits.

The circuit replicates the input-output characteristics of the gradient detector (Figure 5.8). The processes carried out in the neurons are replicated in a finite state machine, the internal state of which changes according to the pulses present at the inputs. To distinguish the digital implementation from the neural circuit, its inputs are renamed as shown above. Input A corresponds to neuron N1 and input B represents neuron N2. Yet, to maintain the relationship with the neural architecture this micro circuit is called „2-before-1“ in the subsequent sections. Let us assume that the state of the D flip-flop is set to „1“ after the initial system reset. Then, pulses on input A will pass the AND-Gate and appear on the output side. The state of the D flip-flop stays at „1“. If, however, input B receives a pulse the state of the D flip-flop is reset to „0“ and remains „0“ until input A receives a pulse sometimes later. Then, this particular pulse on input A is not able to pass the AND-Gate but sets the D flip-flop to its initial state „1“. As long as the state of the D flip-flop

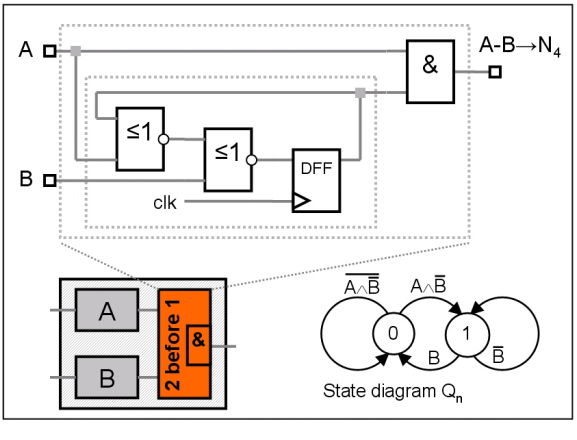


Fig. 15.6 State diagram and circuit for computation of the pulse difference

remains „1“ subsequent pulses on input A can pass the AND gate again until input B receives a pulse and so forth. Thus, the input-output behaviour of this circuit compares to the gradient detector shown in Figure 5.8.

It is important for the correct functioning of the circuit that the pulse duration at the inputs is equal to a clock period of the D flip-flop. This can be ensured through appropriate technical measures in the realisation of the pulse router network. In a topology comparable to Figure 15.9 and through a suitable configuration of the NPU, this state machine can be integrated in a network for feature extraction.

Pulse-Router

The distribution and retransmission of pulse events is based on the usage of an assignment table [96]. At first, the pulses of a certain NPU at coordinates $(x, y)_{src}$ targeted for routing pass the AER coding and a FIFO memory. Subsequently, the associated address in the assignment table is queried, and the corresponding target neurons $(x, y)_{dest1} \ldots (x, y)_{destN}$, $(N \leq 32)$ are identified. In addition, the input (A,B or C, cf. Figure 15.5) of the neuron is identified which processes the pulse (neuron N_1 or N_2 in Figure 5.8 or the accumulator, comparable to neuron 5 in Figure 6.2a). Afterwards, the pulses are distributed to the NPUs of the matrix via a 1-of-128 decoder according to the content of the assignment table (Figure 15.7) . The decoder is clocked at the same frequency as the AER coding; thus it can retransmit at the same data rate across the matrix as captured by the AER, approx. $(200) \cdot 10^6$ pulse events per second.

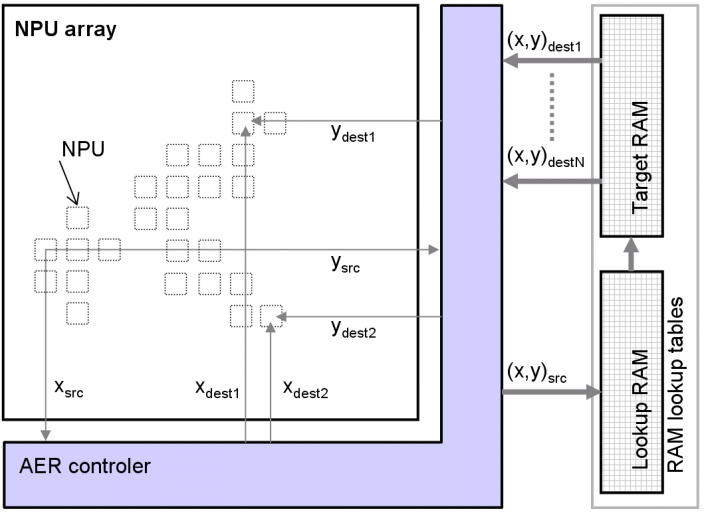


Fig. 15.7 Combination of AER coding and lookup-based routing

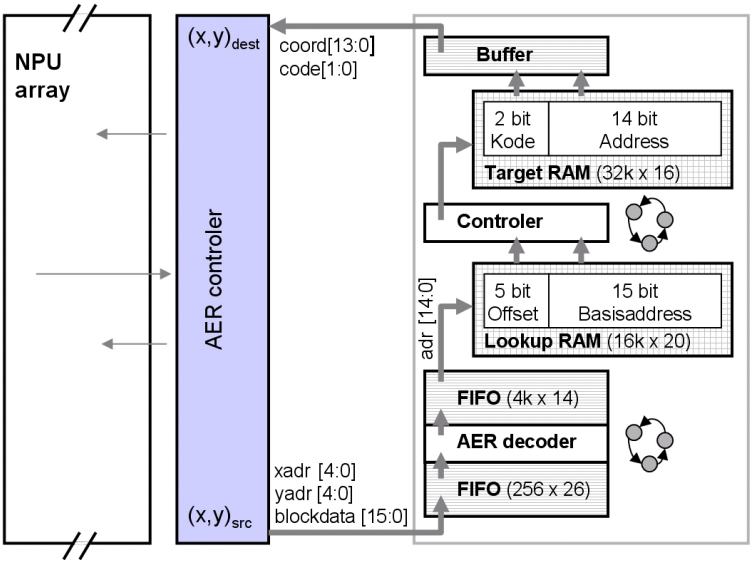


Fig. 15.8 Functional description of the routing of pulse events

The memory organisation of the assignment table is shown in more detail in Figure 15.8. The memory is split in two parts, the first one being attached to single NPUs and containing an address for the second one. This so-called target memory contains a (variable) number of targets for the incoming pulse.

In the first stage, implemented by a Lookup RAM, the pulses arriving from the AER are assigned with a base address and offset for the target RAM. This address denotes the start address in the target RAM where the targets for the incoming pulse are located, while the corresponding address space (i.e. the range of targets) is defined by the offset. Therefore, an incoming pulse can have up to 32 target NPUs (equal to 5 bit offset). The content of the target RAM is divided into a target address, which contains the address of the target NPU ($2 \cdot 7$ bit, in a 128×128 array), and a 2 bit sub-address that defines the input (A,B,C) of the NPU where the pulse is transmitted to. Neurons N_1 and N_2, resp., are represented by A and B, resp., and neuron N_5 by input C. Thus, the target RAM is apportioned in a variable way according to the net topology to be realised. The output pulses are distributed via multiplexers across the NPU matrix according to their target address. This way, the combination of AER and router implements virtual axons/dendrites which can be used to connect the NPUs in complex networks. In addition, external pulses can be entered into the router in parallel with the ones from the AER, so that during testing, external stimuli can be fed to the router IC.

The architecture of the pulse capture (AER) and the pulse distribution (router) is designed for a rate of $200 \cdot 10^6$ pulses/s at $200\,MHz$ system clock.

If a mean pulse rate of $10\,kHz$ for single neurons is assumed, the whole matrix generates an average of $160 \cdot 10^6$ pulses/s. If partially synchronized activity occurs, higher peak pulse rates are possibly in a given time slot or at a certain location. The locality is incorporated in the AER coding, where events of a 4×4 block of the matrix are encoded in a single code word (section 14.2). The length of pulse delay caused by the arbiter varies between $10\,ns$ (2 clock cycles) and $3\,\mu s$. The routing architecture contains FIFOs (Figure 15.8) which enable an intermediate storage of pulse events. The memory depth of the FIFOs determines the largest possible delay of a pulse. The FIFO receiving the AER-coded event has 256 slots and the FIFO for buffering the decoded source addresses contains $4k$ entries/slots, resulting in a maximum delay of $1.26\,\mu s$ and $20\,\mu s$, resp. The minimum delay of the routing architecture is $30\,ns$ (6 clock cycles). System simulations with relevant configurations and topologies have shown that in the majority of cases, the total delay of the pulse transmission (AER and router) is significantly below the pulse duration of a neuron ($1\,\mu s$). Thus, the simultaneousness is approximated well.

The routing as described above is constant, i.e. a fixed configuration always delivers pulse events to the same targets, independently of the activity along the virtual dendrites/axons. However, the routing memory can be reconfigured at runtime.

15.3 Demonstration of the Feature Detection

The feature detectors are especially suitable for the construction of detector profiles for Gabor wavelets. These can be realised in several ways. The starting point is the distribution of input pairs according to the given detector profile. The choice of pixels accessed by N_1 resp. N_2 of Figure 5.8 can be carried out similar to the *Stochastic Universal Sampling* method described in [98].

As shown in Figure 6.10, this results in a convolution mask close to the original Gabor wavelet. When the input A of one or several pulse difference circuits („2-before-1“) access the same image coordinates, the positive filter coefficients of the convolution mask are generated in a number of discrete steps equivalent to the number of parallel accesses. B inputs accessing the same pixel will generate the negative coefficients. After assigning the pixel coordinates of the receptive fields as described above to the inputs of the pulse difference counters, their outputs are merged via a neuron working as a summation element resp. via a weighted accumulator. The output pulses of this summation neuron represent the convolution answer to the input image.

The convolution masks obtained after the distribution process show a strong similarity to the receptive fields found in the mammalian cortex [4]. As mentioned previously, the feature detectors have to cover the whole retina, i.e. a complete set of neurons is required for every Gabor wavelet at a new image location. Depending on the feature and angle resolution necessary, between

20 and 40 different types of Gabor wavelets have to be computed at every location. Also, the feature detectors have to be repeated in a type-specific pixel raster (depending on the image resolution). In order to reduce this overhead, a hierarchical schema for connecting the building blocks may be advantageous, where neighbouring Gabor wavelets use the same subblock for their respective convolution masks. The overhead in connection density incurred by this hierarchical schema has to be weighed against the one of simply copying the convolution mask. The hierarchical view of feature composition is supported by the current view of neurobiological processing, where receptive fields of progressing complexity are built from receptive fields of less complexity derived in earlier processing layers [99]. If the whole image area has to be covered in Gabor wavelets in a 1-2 pixel raster (e.g. for object localisation), overlapping parts of the convolution masks should be reused. This way, an area-saving usage of the pulse difference counters contained in the NPUs is ensured.

Figure 15.9 shows two variants of a hierarchical structure for Gabor feature composition. Depicted are the pulse sources (pixel cells) as well as the pulse difference and accumulator circuits of the NPUs which form the submasks as shown, encompassing 5 neighboring pixels ($x_{n-2} \ldots x_{n+2}$). These submasks with their coefficients $(+1, 0, -2, 0, +1)$ and $(-1, 0, +2, 0, -1)$, resp., form the basis of the complete hierarchical convolution mask, as illustrated in Figure 15.10.

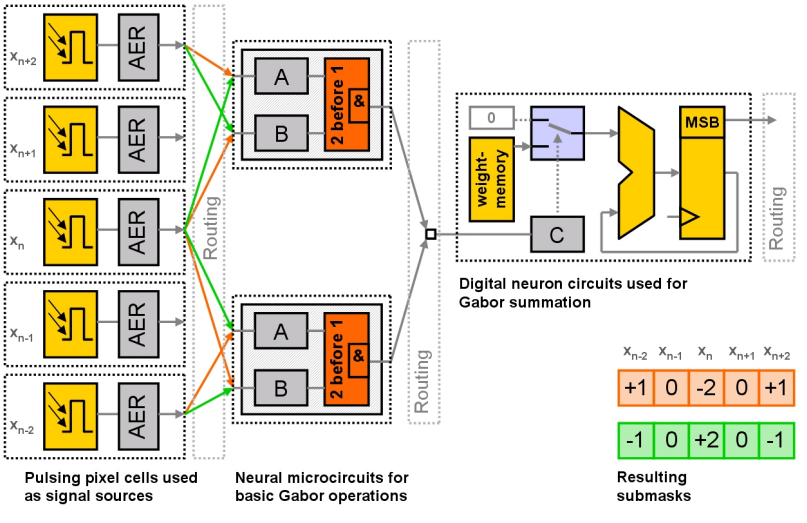


Fig. 15.9 Configuration and NPU topology for feature extraction. The submasks in the lower right corner correspond to the connections of the pulsing pixel cells to the pulse difference counters as shown in the upper right (red resp. green connections)

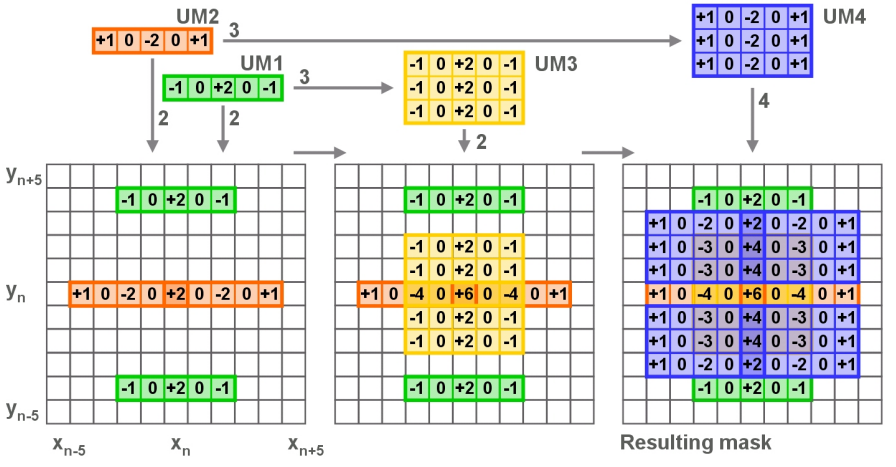


Fig. 15.10 Hierarchical construction of convolution masks from submasks. The upper part of the graphic shows the realisation of more complex masks from the submasks of Figure 15.9. In the lower part of the illustration, the complete convolution mask representing a Gabor wavelet is depicted, resulting from a superposition of all submasks.

To achieve this, further submasks (UM3, UM4) are generated from neighbouring submasks (UM1, UM2) and combined to form a complete convolution mask. The mask coefficients are identical to the number of accessing A (positive) or B (negative) inputs at a certain mask coordinate. The centre of the convolution mask is denoted by (x_n, y_n). The resulting convolution mask is similar to the one shown in Figure 6.10, with differing extension and wavelength.

Construction of the Complete Convolution Using Gabor Wavelets

Figure 15.11 employs the example of Figure 15.1 for a system simulation of the neural processor. The complete processing pyramid has been realised with the NPUs of the router matrix, which have been connected in a hierarchy via the pulse transmission and distribution of Figure 15.8. The figure shows a greyscale representation of the pulse histogram of all NPUs, the processing activity of which can be derived from the relative frequency of output pulses, i.e. the brightness of the pixel representing the NPU.

In more detail, the pulsing pixels cells in the upper left corner are supplied with an image, feeding their output in turn to the first level of NPUs (configured for pulse difference computation, in the space between the colored rectangles). The pulse difference outputs of this step are then added up to form the submasks UM1-4 as shown in Figure 15.10. The submask output is again added up to realise a positive and negative wavelet profile (neuron

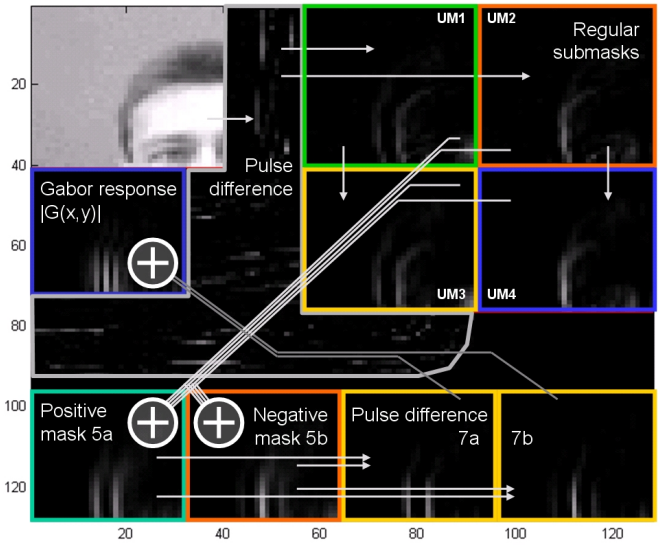


Fig. 15.11 Histogram and signal flow for a complete hierarchical convolution using Gabor wavelets in the NPU matrix of the router-IC. The input image is situated in the upper left corner, 32×32 pixels plus a 6 pixel boundary to cover the extent of the convolution mask. It is covered with 1024 identical convolution masks in a one pixel raster. The output pulses of the NPUs are shown as greyscale values centered on the specific NPU, i.e. a bright image point corresponds to a very active NPU. The NPUs in the upper left corner have their pulsing pixel cells activated, and these are supplied with a image-equivalent photo current. In the rest of the matrix, the NPUs are configured according to their specific function in the convolution hierarchy as pulse difference counter („2-before-1“) or as summation element (also cf. Figure 15.9).

N_{5a}, N_{5b}). The rectangle „positive mask“ is thus equivalent to the convolution of the partial image with the Gabor wavelet resulting from Figure 15.10. The same submasks are also used to derive a Gabor wavelet with the opposite characteristics, i.e. negative coefficients („negative mask“). Accordingly, a symmetrical convolution mask response comparable to neurons N_{5a} and N_{5b}, resp., of Figure 6.2 is computed. From the convolution answer to both masks, a median-free convolution answer is derived in a two-way difference computation („pulse difference“), as in N_{7a} and N_{7b}, resp., of (Figure 6.8). The last step comprises the summation of these parts of the Gabor wavelet to derive the absolute value response (illustrated in the left half of Figure 15.11 below the original image).

16384 NPUs are sufficient to cover a quarter of the original image completely with the convolution mask corresponding to a single feature (phase, extension, wavelength and orientation). The pulsing pixel cells operate at a

mean pulse frequency of 3 kHz, i.e. a detector answer at a resolution of 256 greyscale levels can be obtained after about 85 ms. To achieve a higher resolution of max. 60 dB, up to 10^3 single pulses can be added up for a detector response.

A filtering of the whole image would need 4 chips. On the other hand, for a contiguous area as in Figure 7.2, 960 real-valued Gabor detectors are needed, which would use about 90 % of the resources of a single router IC (Figure 15.11). Consequently, 6 chips are necessary for realising the filtering of both eye regions and the mouth with two different Gabor filters each. This figure includes a certain overhead for peripheral processing functions.

15.4 Summary

This chapter has described a system concept which is able to realise arbitrary pulse based convolution by using special digital neural computation elements (Neural Processing Units) in suitable network topologies. Thus, an integrated circuit has been developed which incorporates the basic neural functions necessary for feature detection (pulse counter, weighted accumulator) as well as facilities for pulse transmission and distribution. In addition, image sensors enabled for a pulse-based interface have been integrated on the chip.

The chip contains the basic building block of a feature cascade which is optimal also for a 3D stack. The adapted version of this chip, with improved utilization of the NPU matrix, is described in chapter 17.

16

3D Stacking Technology

Arne Heittmann

The preceding chapters presented fundamentals of pulsed neural networks for image processing. The outcome was a modular architecture containing dedicated layers for each subtask. Finally, it was shown how dedicated VLSI circuits supporting these subtasks can be implemented. The integration of particular components (CMOS imager, feature cascade, object recogniser) into a system poses further questions, however. Particularly, the problem of interconnection of neurons and neural nets was only discussed in a rudimentary way. This chapter is devoted to associated questions and presents the first results and solutions.

Let us assume an imager with a resolution of 128×128 pixels. As we have shown in the previous chapter the extraction of a particular orientation using a Gabor wavelet requires a dedicated detector chip in a $130\,nm$ CMOS technology. A state-of-the-art $40\,nm$ CMOS technology would permit a realisation of 8 detectors of different orientations on a single chip. Commonly, 20 to 40 complex-valued wavelets are used in state-of-the-art image recognition systems. Hence, an overall number of 6 to 12 chips are required for feature extraction only.

Note that the required bandwidth increases with

- the resolution of the image sensor,
- colors to be processed,
- increasing number of features (more orientations, different scales),
- increasing number of objects to be recognised.

If we consider further that those networks are not only useful for the extraction of features within images, but also for detection of features within data in general, it becomes obvious that the overall number of chips used for feature detection would blow out.

A hardware implementation can only be realised in a reasonable and low-cost way if a single programmable device for feature detection is made available, and which can be used in a replicable way within the architecture. Since

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 269–298, 2010.
© Springer-Verlag Berlin Heidelberg 2010

in the above case more than 160×10^6 events per second have to be exchanged among different modules, minimal interconnections lengths, an optimised memory architecture, as well as a powerful bus architecture are mandatory. Then, real time capabilities as well as a reasonable amount of power consumption can be guaranteed. Bus architectures capable of efficient data transmission were discussed in chapter 15. This chapter is devoted to issues of assembly and system integration.

Today's state-of-the-art packaging technologies offer an overall pincount of several hundreds pins and provide a limited degree of parallelism for communication [100]. As a consequence, communication between modules appears to be serial-like, with high clock frequencies if standard packaging technologies are chosen for system integration. Especially for pulse-coded information processing, this fact poses particular problems for data transmission. For our application, pulse events must be detected on the sender side, must be encoded for transmission and must be reconstructed on the receiver side. The delay between the onset of an event on the sender side and the reconstructed pulse on the receiver side has to be as small as possible in order to keep up the correlation of pulses pairwise. In particular, let us consider two synchronised pulses of two arbitrary neurons on the sender side. If these pulses are transmitted in series by means of a serial interface to a receiver, a misalignment of the reconstructed pulses occurs. The correlation of both pulses, which potentially carries information, degrades to a certain degree. This degradation can be minimized if transmission rates are raised to sufficiently large values. For instance, if pulse frequencies are in the range of $100\,kHz$, the transmission units operate with $100\,MHz$. The second problem with standard packaging technologies is directly linked to power consumption. In order to minimize both degradation of correlation and power consumption, we have to realise minimal distances between a sender-receiver pair. In the same way we have to minimize the distance between particular chips. Unfortunately, the state-of-the-art technology for assembly and system integration severely limits these demands.

In our case, power consumption is determined by the communication architecture. If we look at pulsed coded neural networks, encoding circuits as well as decoding circuits operate with high clock frequencies. The power consumption of these modules accounts for more than 50% of the total power consumption (cf. chapter 15). Long communication paths cause contributions to dynamic power consumption due to large wire capacitances. Supply pins as well as supply wires have to be designed with respect to current densities and I·R drop. Additionally, the impact of electromigration, as well as generation of heat on the system reliability, has to be taken into account. Eventually, cooling equipment has to be mounted on the chip, which causes additional system costs.

Besides power consumption, the physical volume of a system plays an essential role for a vision system. Realizing a complete vision system using standard technology would lead to a bulky system containing many PCBs, slots

and space for wiring. This is acceptable for an experimental setup. However, for mobile and battery-driven platforms like robots, it is not. Furthermore, the reliability of such a system with hundreds of chips poses a problem not to be neglected.

16.1 Fundamentals of 3D Stacking

In order to meet individual requirements concerning reliability, volume, communication bandwidth and power consumption, a novel 3D chip stacking technology was developed. The proposed technology allows us to stack and connect several chips in a vertical direction very efficiently. The electrical and mechanical bonding is realised through the chip surface, whereas the electrical connections serve as mechanical bonds.

In Figure 16.1 the procedure of 3D stacking is illustrated. On the left hand side the standard scheme for system integration is outlined, chips will be 2D-mounted on a shared supporting substrate. The shared substrate on the one hand serves as a mechanical support, on the other hand electrical interconnections as well as supply lines are integrated on the board. Figure 16.1b illustrates an alternative way of system integration. Chips are vertically stacked. Using this structural shape results in less area occupation as well as less allocation of physical volume. Both assembly techniques can be combined (Figure 16.1)c.

One of the fundamental advantages of the 3D stacking technology is that wiring lengths of interchip connections are drastically reduced to a factor between 100 and 1000 (Figure 16.2).

Additional to the standard connection scheme via the chip periphery (e.g. chip border), 3D technology provides an opportunity to electrically connect a chip via the surfaces (i.e. the front side surface and the back side surface). These contacts allow direct connections to an adjacent layer of the stack with-

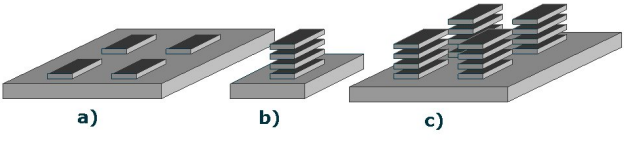


Fig. 16.1 (a) Outline of two-dimensional system integration. Individual chips were mounted on a supporting substrate (i.e. PCB) (b) Outline of 3D-chip integration. Individual chips of the system are vertically stacked and both electrically as well as mechanically connected. Compared to (a) a considerable reduction in the area occupied shows up. (c) Combination of (a) and (b) on a shared supporting substrate.

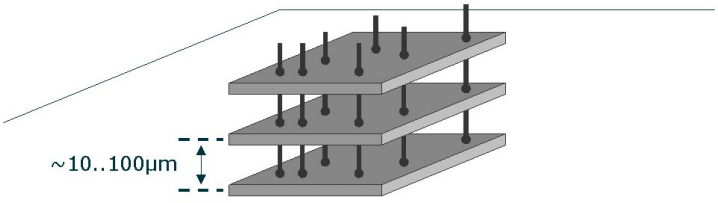


Fig. 16.2 In 3D chip stacks the interconnection length can be downsized to $10\,\mu m$.

out any additional bond wires. 3D packaging technology offers very short interconnection lengths. But the overwhelming benefit is the increased connection density: the increased number of connections permits the parallel transmission of signals in-between adjacent layers of the chip stack. Architectures of pulsed neural networks significantly benefit from parallel data transmission. High performance coding circuits connected to a transmission channel with few signals can be replaced by much simpler circuits which operate in parallel. The clock frequency can be reduced since the required bandwidth for communication now spreads over many parallel signals (cf. chapter 17). The usage of shorter interconnection lengths results in reduced parasitic capacitances and reduced switching power, and the overall power consumption will be reduced.

Interconnection lengths are given by the required distances between chip surfaces as well as the layer thickness of the stacked chips. As explained later, thinned substrates are applied to chip stacking, consequently interconnection lengths are in the range of a few $10\,\mu m$. Hence, densely connected systems can be realised in a space-saving way.

In a first approximation, electrical 3D connections can be modelled as simple RC elements. The signal delay is given by the Elmore delay [101]:

$$\tau_{Via} = \frac{1}{2} \cdot R' \cdot C' \cdot L^2 \ . \tag{16.1}$$

The parameter L represents the distance from a bottom solder pad to the location where the signal terminates. The parameter R' models the resistance per unit length, likewise C' models the capacitance per unit length (Figure 16.3).

If a typical cross sectional area of $30\,\mu m^2$ is assumed for a vertical via, the magnitude of resistance can be estimated to $5\,m\Omega$. This estimation is based on ideal parameters and represents a lower bound. From practical measurements of resistances, less than $1\,\Omega$ were obtained [102].

Likewise, capacitances less than 100 fF were measured [103]. Taking both results into account, we can expect time constants within the range

$$0.1\,\text{fs} < \tau_{Via} < 100\,\text{fs} \tag{16.2}$$

from one chip layer to the next.

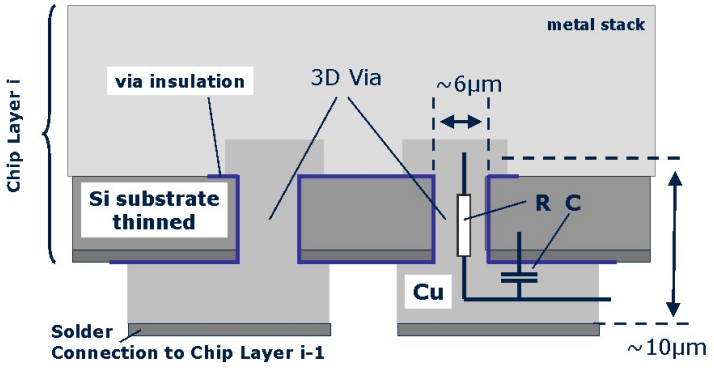


Fig. 16.3 Cross section through a vertial interconnection

Concerning performance issues, it follows that the design of a communication architecture is much simplified compared to a standard communication scheme. Firstly, it is possible to connect local signals in a direct way from one layer to the next without the need to route local signals over the chip boundary. The influence of skew, jitter as well as crosstalk on the signal integrity will be significantly reduced. Secondly, due to the multiplicity of connections, the bandwidth is significantly increased. Hence, circuits for signal transmission benefit from 3D stacking, since the same amount of information can be transmitted with reduced clock frequency.

From the technological point of view some challenges arise in the provision of a robust 3D stacking technology.

- realisation of high density 3D connections,
- etching of 3D vias with high aspect ratio,
- exclusive usage of processes having a low temperature budget,
- preparation of low-temperature processes for via metallisation,
- wafer thinning and handling of thinned wafers,
- precise alignment of chips during assembly process,
- efficient test strategies,
- low production costs compared to classical solutions.

Subsequently, a 3D stacking technology will be presented which meets almost all of these demands.

16.2 Processing Steps for 3D Stacking

At present several different approaches for 3D stacking technology are proposed which differ with respect to technological effort as well as to electrical properties of the interconnections. The demand for high contact densities and an unlimited number of stackable layers precludes particular processing steps

and requires a co-ordinated processing chain at particular stages of the production flow.

Electrical connections have to be provided at the front side as well as from the back side into the active area of stacked chips. Special attention has to be paid to the processing of the contacts on the back side. Vias have to be etched deeply into the substrate while each via is characterised by a small cross sectional area. Since metallisation of these vias only succeeds up to a given aspect ratio (ratio between depth and diameter of the via) the depth of the vias has to be as small as possible, i.e. at some stage of the processing only thinned wafers can be used as a chip substrate. Therefore, thinning of wafers as well as handling of thinned wafers is a special problem posed by 3D stacking technology.

Primarily, chips originate from standard CMOS production. In order to make use of an almost unchanged CMOS process, all additional processing steps for 3D are made in the so-called back end, i.e. after CMOS has been processed. There are different advantages in retaining standard CMOS. Firstly, the electrical re-characterisation of the complete process can be almost omitted. Most parameters for design and simulation can be re-used. Existing designs can also be re-used in a 3D stack without major changes. Secondly, re-centering of CMOS processes due to impacts caused by 3D processing steps can be also omitted. The fundamental advantage of this kind of processing is that only processes with a low temperature budget up to $400°C$ are of use in the back end. Due to this constraint, the yield is much better as compared with other processes.

The stacking method presented is divided into particular processing steps as given in Figure 16.4. All processing steps can be implemented preferably on standard equipment by way of batch processes until the assembly completes the processing chain. The mechanical bonding is realised as a chip-to-wafer bonding process. The base substrate is given by a processed wafer. Before the assembly process, all chips will be tested on-wafer and tagged with respect to faultless functionality. Afterwards, wafers are split into single chips.

During the assembly process only fully functional chips (known-good-dies, KGD) will be bonded in a successive way. In contrast to a wafer-to-wafer process this approach has several advantages:

- improvment of yield,
- chips having different geometric shapes can be bonded without wasting wafer area,
- different thermal expansion coefficients of substrate as well as of particular layers of the stack can be neglected,
- inhomogeneities of the wafer surface and the chip surfaces can be neglected.

A three-layer stack consisting of four chips is shown in Figure 16.5. The base substrate has to have a slightly larger footprint than the stacked chips in order to add bonding wires at the chip periphery, and unlike the stacked chips, the base substrate doesn't have to be thinned. From the outline given in Figure 16.5, it is seen that 3D vias are running through the thinned substrate

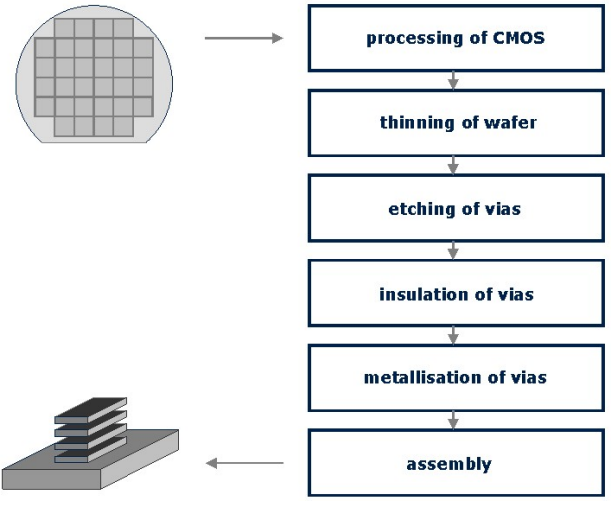


Fig. 16.4 Overview about particular processing steps for 3D

and terminate on the first metal layer of CMOS. Unlike within other 3D approaches, it is not necessary to realise vias running through the entire chip from the back to the front. The standard back end of CMOS already provides a metal stack which can be used to connect to the front side of the chip. In principle, with the proposed concept it is possible to realise high density vias having less cross sectional area compared to vias running through the entire chip. The length of the via compares to the residual thickness of the thinned substrate (cf. Figure 16.3).

In the following paragraphs individual processing steps will be presented in detail.

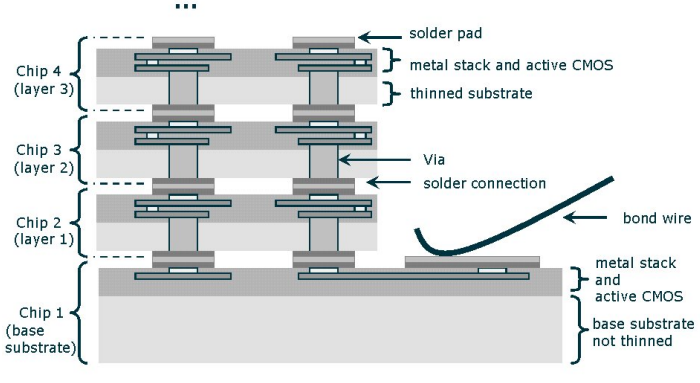


Fig. 16.5 Cross sectional view of a 3 layer stack (4 chips)

Thinning of Wafers

As outlined in the foregoing chapters, thinning of chip substrates is necessary if the focus is on the realisation of high-density contacts. A second challenge arises from the fact that all 3D processes are back end processes, i.e. processes that will be applied after CMOS has been finished. Since the complete metal stack of CMOS exists, front side contacts can be realised almost effortless, for only the front side passivation has to be opened at particular positions, which is already part of the standard process. The realisation of back side contacts is more complicated. Here, special back side processes have to be applied. It is particularly challenging to align the first mask with the CMOS structures so as to achieve a precision better than 10μ. Well-established methods like infrared light suffer from low precision worse than 10μ. In addition, the metal stack absorbs a substantial fraction of the radiation.

Our approach exploits purpose-built etch markers which will be inserted on the wafer's surface prior to CMOS processing. Typically, these markers will be realised using a dry etching step on the front side of the raw wafer. The groove-like markers provide a sufficient contrast for short-wave radiation which is essential for a high-precision alignment of masks with the target wafer. Both the first mask for CMOS, as well as the first mask of the back side back end, will be aligned to theses markers.

Another challenge is the precise control of residual wafer thickness during the wafer thinning process. It is of fundamental importance to have knowledge about the residual thickness, since parameters of the via etch process strongly rely on the requisite via depth to avoid undercutting as well as overetching. Precise control of residual thickness is realised using a shallow interlayer which is not susceptible to particular chemical etching solutions. Here, this layer consists of an epitactically grown boron-doped silicon deposit having a doping which varies between $8 \cdot 10^{19} cm^{-3}$ and $1 \cdot 10^{20} cm^{-3}$. The thickness of the layer is about $750\,nm$. If a KOH solution is used for wafer thinning, the etching rate drops by two orders of magnitude as soon as the doped interlayer is exposed to the etching solution. After etching, the back side of the wafer appears to be an outstanding plain surface which is well-suited for successive processing steps. After the interlayer has been deposited, an epitactical silicon layer with a regular doping profile is grown on top of the interlayer. The thickness of this silicon layer equals the residual thickness of the wafer after thinning. Both uniform-doped layers having a doping of $1 \cdot 10^{15} cm^{-3}$ and heavily-doped layers having a doping of $1 \cdot 10^{16} cm^{-3}$ can be realised.

The thinning process is explained in Figure 16.6. Special attention should be paid to the topology of the wafer's surface after the second epitactical layer has been grown. As we can see, the topology of the marker is replicated on the surface of the epitactical layer (Figure 16.6(c)).

In Figure 16.7 the topology of a prepared wafer before as well as after the deposition of an epitactical layer is shown. The growth of crystalline silicon is controlled mainly by the crystallographic orientation of the substrate. At the

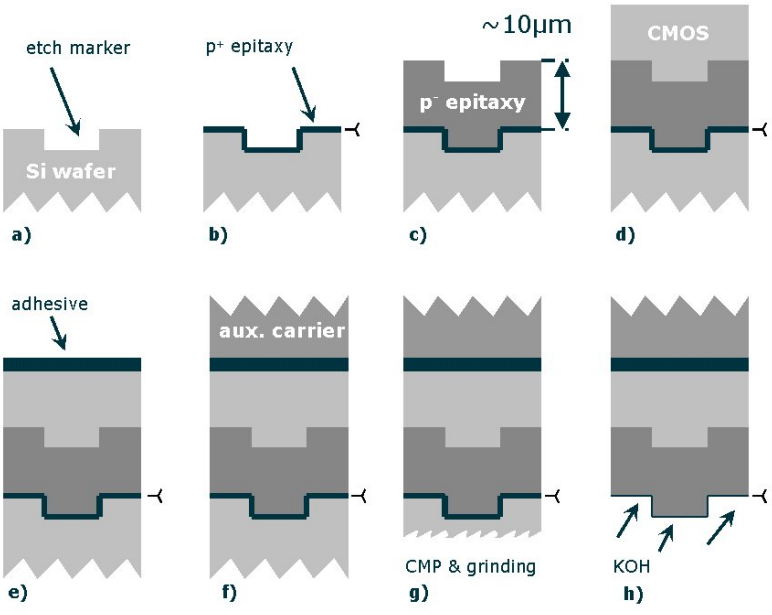
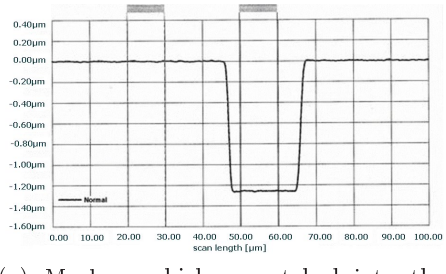
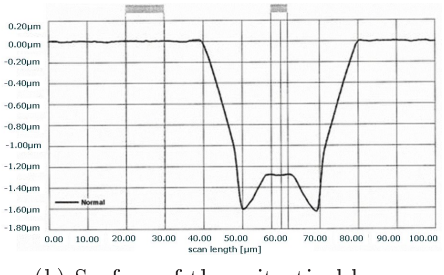


Fig. 16.6 Preparation of wafers and wafer thinning. a) Etching of markers for the purpose of mask alignment. b) Deposit of a p^+ epitactical layer. c) Deposit of epitactical layer used for standard CMOS. d) CMOS processing. e,f) Attachment of an auxiliary carrier in order to stabilise the substrate during the thinning process as well as during back side processing. g) Mechanical thinning of the wafer's back side and CMP (chemical mechanical polishing) h) Thinning using a KOH solution, thinning process stops at the interlayer (the sentinel points to the position of the p^+ layer).



(a) Markers which are etched into the front side surface of the raw wafer



(b) Surface of the epitactical layer

Fig. 16.7 Topology of the surface nearby the markers; measurements after H. Schäfer, Infineon Technologies AG, Munich

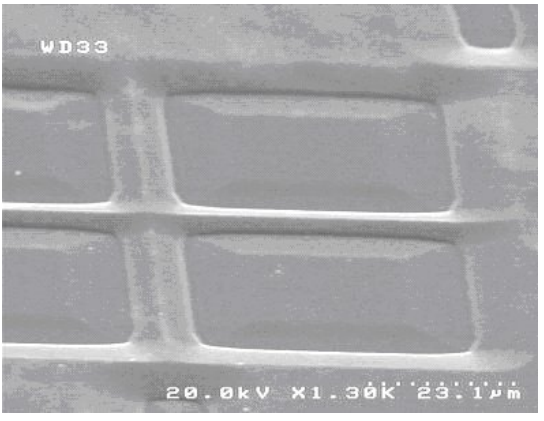


Fig. 16.8 Characteristic structure on the wafer surface for the purpose of mask alignment

edges, the process of growth lets the topology spread. At the lower edges, the rate of growth is reduced compared to the inner area of the etched groove, which results in different sizes of layer thickness. Near the edges, the layer thickness drops below the default thickness.

In Figure 16.8 a REM photograph of a typical marker structure is shown. The back side processes as well as front side processes can be mutually aligned by means of these markers.

After CMOS processing and prior to the thinning process an auxiliary carrier is attached to the wafer via a polyimide-photoresistant compound (Figure 16.6e,f). This auxiliary carrier stabilises the thinned substrate and remains attached until the assembly of the particular chip is done.

The wafer thinning is carried out in three steps. First, the wafer is thinned using a mechanical thinning procedure (e.g. grinding or lapping). At present, grinding is a standard practice for chip fabrication. During this step the thickness of wafers with an initial dimension of $600\,\mu m$ will be reduced to a residual thickness between $30\,\mu m$ and $40\,\mu m$. Typically, the surface roughness is better than $1.0\,\mu m$ using standard equipment for grinding. Next, the remaining defects of the surface will be removed, utilizing a CMP step (chemical mechanical polishing). Similarly to grinding, CMP is a standard procedure for chip fabrication and is usually adopted for surface planarisation. Common CMP solutions contain 4 % particles of SiO_2 having a size of $10\,nm$ [104] as well as KOH which has been dissolved in water. The pH value of the KOH solution typically ranges from 10 up to 11 [105].

Finally, step 3 of the thinning process includes spin etching, using an etching solution with 10 % KOH dissolved in water. Figure 16.9 illustrates the back side structure of a prepared wafer after etching with KOH. The uniformity of the marker structures, as well as the quality of the surface, is visible.

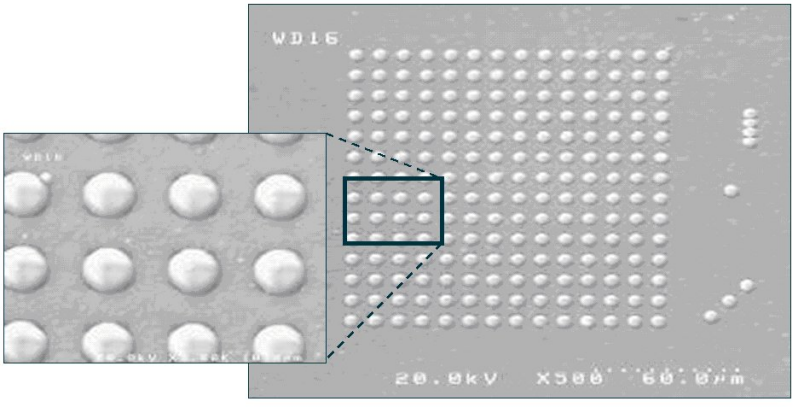


Fig. 16.9 Back side topology of a sample after etching using KOH

Etching of Vias

If large contact densities are aimed for, the via holes have to be characterised by a small cross sectional area, a large via depth and almost perpendicular side walls. Hence, in order to fulfil these essential demands, the etching process for the vias has to feature an excellent anisotropy. The so-called DRIE process (also noted as ASE process, Bosch process) proved to be especially advantageous for etching of via holes [106]). In Figure 16.10 the basic operation is shown. The process is an alternating sequence of etching and passivation. The number as well as the duration of each iteration determines the roughness of the side walls and the depth of the via.

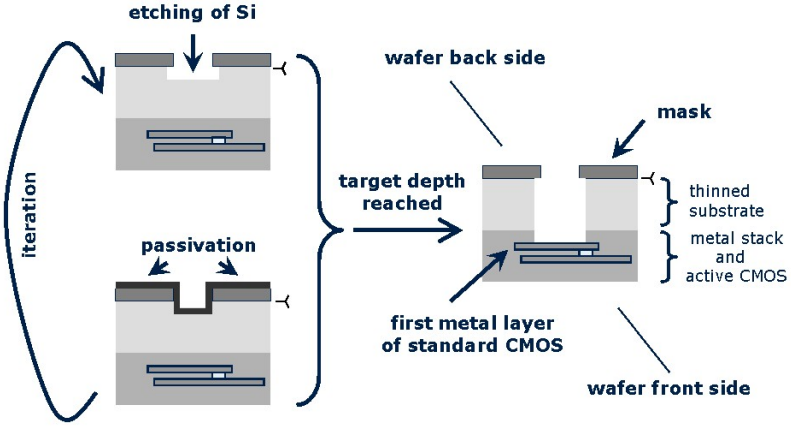


Fig. 16.10 Deep Reactive Ion Etching (DRIE)

When the residual wafer thickness matches the via depth given by the DRIE process the vias will terminate at the first metal layer of CMOS. It is now possible to establish an electrical connection from the backside to CMOS.

Etching and passivation alternate. The etching is performed in a reaction chamber using a inductively coupled plasma. The etching is initiated after a SF_6 gas has been introduced into the chamber. Free flourine radicals are generated from sulphur-hexaflouride which react on the silicon surface towards volatile flourine-silicon compounds. Passivation is done by introducing gaseous $C_xH_yF_z$ compounds (e.g. CHF_3, C_4F_8) into the reaction chamber. By choosing appropriate process parameters (e.g. pressure, temperature) polymerisation of carbon and flour begins. The nCF_2 polymer deposits on the side walls of the vias and acts as a protective layer during the following etching step. In this way vias having an aspect ratio up to 30 : 1 can be realised. The roughness of the side walls is sufficiently small which helps during the filling process. Within one iteration typical etching rates are in the range of $1 - 2\,\mu m/min$.

In Figure 16.11 an etched via structure is shown after the procedure given in Figure 16.10 has been finished. Perpendicular side walls are clearly seen, which demonstrates the outstanding anisotropy of the etching process.

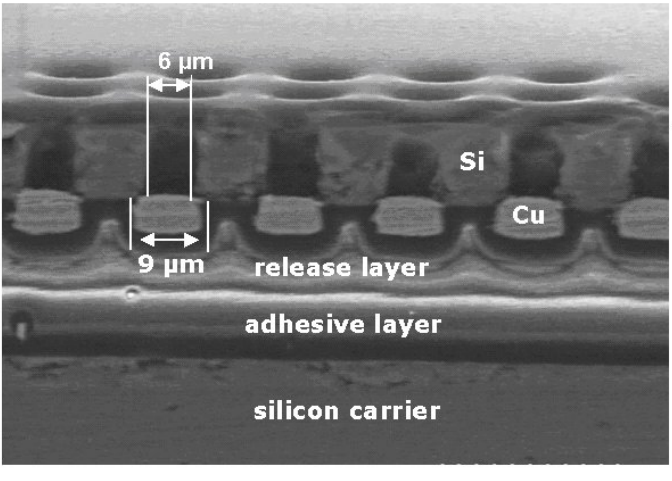


Fig. 16.11 Cross sectional view of a vertical via structure which has been integrated into a thinned Si substrate. The thinned substrate is attached to a carrier using two adhesive layers. The carrier substrate stabilises the superstructure. The back side of the substrate was structured using a SiO_2 mask. The mask defines the locations of the particular vias. Below the etched via structure copper lines are located which act as so-called landing-pads. The presented view shows the superstructure after sawing.

Insulation of Vias

SiO_2 applies particularly to the electrical insulation of vias. The back side of the wafer can be electrically insulated, too. Finally, SiO_2 can be used for realising masks. As mentioned earlier, only a low temperature process can be applied to 3D processes. In order to deposit SiO_2 as an insulator, an optimised plasma enhanced CVD-process is used here (a so-called LT-PECVD process, a low temperature, plasma-enhanced chemical vapor deposition).

First, an oxide will be deposited on the back side of the substrate which serves as a mask for the definition of the via locations. This oxide layer has a thickness of $1\,\mu m$ after deposition. After the etching of the via holes, the oxide will be removed along with the oxide which separates the first metal layer of the metal stack from the silicon. The thickness of this oxide is in the range of 600 nm to 800 nm. After etching, the back side oxide has a residual thickness between 100 nm and 200 nm. This layer has a sufficient thickness to reliably insulate contacts on the back side from the Si substrate.

The plasma is mainly used to generate chemically reactive constituents which essentially support the deposition of the oxide. As a side effect, electrons which were generated by the plasma cause the synthesis of additional by-products due to ionisation and dissociation. Helium is passed into the reaction chamber in order to significantly reduce the concentration of undesired by-products.

The reaction equation which describes the deposition of SiO_2 using CVD is given by

$$SiH_4 + 2\,N_2O \rightarrow SiO_2 + 2\,H_2 + 2\,N_2.$$

The reactants are, on the one hand, silane (SiH_4) and, on the other hand, dinitrogen monoxide (N_2O, nitrouse oxide). The optimisation of the deposition process is done by varying the flux of N_2O and the flux of SiH_4 at different levels of temperature. The capacitively coupled RF power was regulated to approximately 20 W while the pressure in the reaction chamber was stabilised at 1 Torr ($0.133 \cdot 10^{-3}$Bar) for all test series.

From Figure 16.12a it can be seen that the rate of SiO_2 deposition increases along with a rising temperature. Likewise, the deposition rate increases if the flux of reactants is increased. If the ratio of flux between N_2O and silane exceeds a specific threshold, the deposition of oxide is limited by the flux of silane. Finally, if oxygen is supplied excessively, a so-called power formation is observed [107] which has to be avoided in general, since layer properties degrade, surface roughness increases and deposition rate decreases.

The quality of the oxide can be assessed in several ways. Typical qualities to be examined are surface roughness, geometry of deposition, the stoichiometry of the oxide and electrical properties of the insulation layers. The surface roughness of the insulation layers is of marginal interest. In accordance with results of the literature [108] it was found experimentally that the roughness of the surface increases if the deposition temperature is lowered.

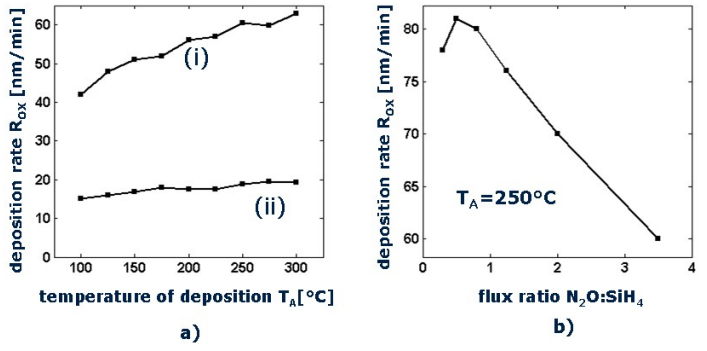


Fig. 16.12 a) Dependency of deposition rate on temperature. (i) Flux of N_2O is 710 *sccm*, flux of SiH_4 is 200 *sccm*. (ii) Flux of N_2O is 177,5 *sccm*, flux of SiH_4 is 50 *sccm*. b) Dependency of deposition rate on the ratio of flux. The overall flux was stabilised to 910 *sccm* . Measurements after [102].

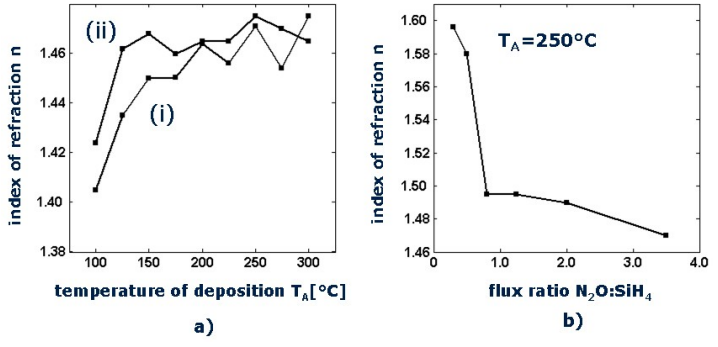


Fig. 16.13 a) Dependency of refractivity on temperature. (i) Flux of N_2O is 710 *sccm*, flux of SiH_4 is 200 *sccm*. (ii) Flux of N_2O is 177,5 *sccm*, flux of SiH_4 is 50 *sccm*. b) Dependency of refractivity on the ratio of flux. The overall flux was stabilised to 910 *sccm*. Measurements after [102].

The stoichiometry gives information about the ratio Si : O within the insulation layer. Variations apart from the ideal ratio may result in impurities which have an impact on the band structure, causing leakage current, for instance. It is possible to get information about the stoichiometry by measurement of the refractive index. Figure 16.13 illustrates the dependency of the refractive index on the deposition temperature as well as on the ratio of flux. Referring to the refractive index $n_{SiO_2} = 1.46$ of pure SiO_2 it emerges that the quality of the oxide becomes better if low temperatures are used and a low ratio of flux is chosen. Equally, raising the fraction of oxygen within the ratio of flux results in a better oxide quality.

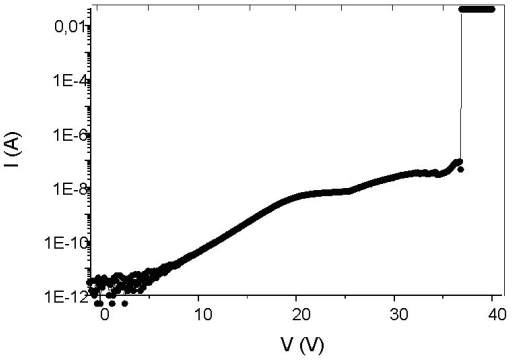


Fig. 16.14 I-V characteristic of a sample having an oxide layer of 50 nm thickness. The layer was deposited on a planar Si surface. The breakthrough voltage was determined to 37.5 V (equivalent to an electric field of 7.7 MV/cm).

The electrical properties of the oxide insulator are of major interest. In Figure 16.14 the I-V characteristics of a prepared layer are shown. Low electrical fields cause a distinct leakage current. In contrast, if the electrical field strength is high enough, a characteristic electrical breakdown is observed.

The current rises quickly with the applied voltage. If the voltage has a magnitude close to the breakdown voltage, the effect of the so-called Fowler-Nordheim tunneling [109] is the dominant effect for current transportation. At lower voltages, current is conducted using impurities and so-called traps (Poole-Frenkel effect [110]).

In Figure 16.15 the dependency of the leakage current on different process parameters is shown. The electric field was fixed to $E = 1\,MV/cm$. In accordance with Figure 16.13, the quality of the insulator is enhanced at a low flux rate of the reactants. Likewise, the quality rises if either more oxygen is supplied or the deposition temperature is higher.

If the temperature of deposition is less than $200°C$ electrical fields larger than $7.5MV/cm$ cause breakthrough current. If the temperature of deposition is higher than $200°C$ breakthrough is observed at electrical fields larger than $9.5\,MV/cm$, which was regarded to be sufficient for the target application.

In Figure 16.16 different possible shapes of oxide deposition are sketched. At slight aspect ratios, the deposition of SiO_2 appears to be uniform (Figure 16.16b), both at the side walls as well as near the landing pad. Near the via opening at the top, the window is narrowing due to a widening oxide layer. This widening is caused by an eased exchange of reactants in the region close to the via opening.

At the least, a layer thickness of one-third is obtained near the via bottom, compared to the thickness of oxid layer at the back side ($1\,\mu m$).

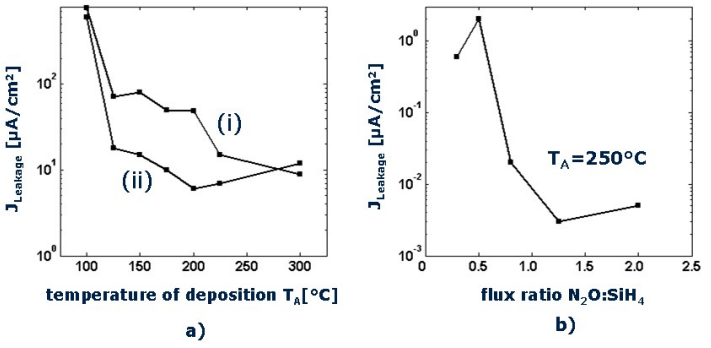


Fig. 16.15 a) Dependency of leakage current on deposition temperature. (i) Flux of N_2O is 710 *sccm*, flux of SiH_4 is 200 *sccm*. (ii) Flux of N_2O is 177,5 *sccm*, flux of SiH_4 is 50 *sccm*. b) Dependency of leakage current on the flux ratio. The overall flux was stabilised to 910 *sccm*. Measurements after [102]. The applied electrical field was $E = 1\,MV/cm$.

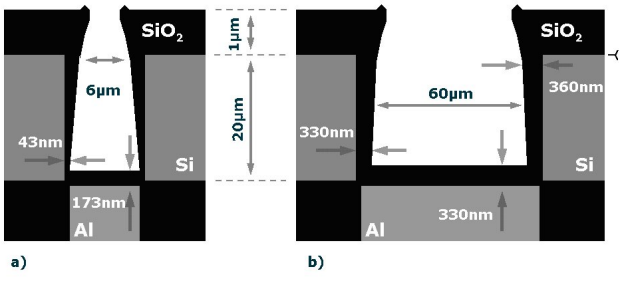


Fig. 16.16 Possible geometrical structures of deposited SiO_2 for vias having different aspect ratios a) $20\,\mu m$:$6\,\mu m$ b) $20\,\mu m$:$60\,\mu m$

The deposition rate of SiO_2 near the landing pad is notably low if the aspect ratio is comparatively large (cf. Figure 16.16a). The maximum deposition rate is obtained near the via opening at the top. During oxidation the via opening is closing more and more, and the exchange of matter into the via hole is more and more limited. Consequently, the thickness of the SiO_2 layer narrows towards the landing pad. The geometrical shape of any usable vias is somewhere in between the extreme cases outlined above, i.e. for all practical use, a sufficient electrical insulation can be guaranteed.

In order to gain information about the resistance of the SiO_2 layer, a measurement setup was used as shown in Figure 16.17b. After the deposition of SiO_2 the via was filled with copper, using an electroplating process. This particular measurement setup was chosen in order to recreate a via close to real conditions found in the applications later. In section „via metallisation“ full particulars about metallisation of vias are given. The layer thickness of

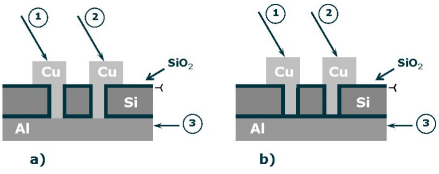


Fig. 16.17 Setup to gain information about electrical properties of vias. a) Measurement of via resistance. b) Measurement of insulator resistance.

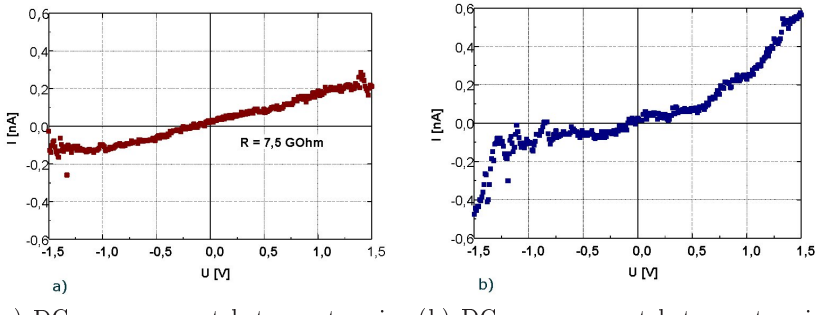
deposited SiO_2 is $250\,nm$. From Figure 16.18 it was found that the leakage current is less than $1\,nA$ if voltages less than $1.5V$ are applied.

For some applications in the analogue domain, this leakage current is still too high. Particular defects of the oxide (so-called pinholes) as well as weak layers aid current transportation. This is especially crucial for the oxide layer close to the landing pad at the side walls of the via (cf. Figure 16.16a).

It is possible to get rid of these defects by applying an anodic oxidation step. A draft of the setup is given in Figure 16.19a. The sample (i.e. a thinned wafer or a particular chip of a thinned wafer) is mounted on a sample carrier. The carrier is designed in such a way that when the carrier holding the sample is immersed into the electrolyte, only the backside of the sample will be coated by the electrolyte. In addition, an electrical connection is established to the p^+ layer of the sample, which acts as the anode.

Since this layer is an excellent conductor, the p^+ layer can be regarded as a equipotential region at low anodic currents. Indeed, the voltage drop at different locations of the p^+ layer to the electrolyte is evenly distributed.

Viable results will be obtained using an electrolyte solution of $0.02\,M$ potassium nitrite and $0.02\,M$ potassium nitrate, both dissolved in ethylene glycol having a proportion of $2\,\%$ residual water [111].



(a) DC measurement between terminals 1 and 2

(b) DC measurement between terminal 1 and 3

Fig. 16.18 I-V characteristics of insulated vias. Results after [102]

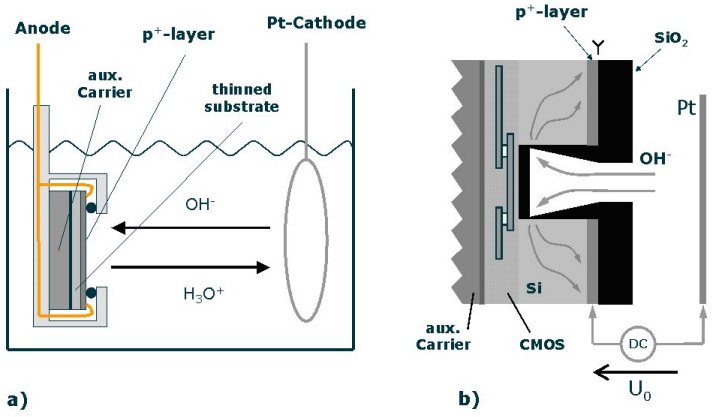


Fig. 16.19 Setup for anodic oxidation. a) Sample, sample carrier and platinum cathode, immerged into electrolyte. b) Closeup of a via

While the solution is being prepared the potassium nitrate dissociates according to:

$$H_2O + KNO_3 \rightarrow K^+ + OH^- + NO_3^- + H^+ \ .$$

The dissociation causes a concentration of OH^- ions in the solution. If the voltage U_0 is applied (cf. Figure 16.19b) OH^- ions migrate towards the substrate due to the emerging electric field. Large current densities emerge at regions where the SiO_2 layers are notably thin, especially at the side walls near the landing pad and at particular defects. The OH^- ions migrate through the SiO_2 layer towards the silicon surface. At the silicon surface the OH^- ions react to SiO_2:

$$Si + 4\,OH^- \rightarrow SiO_2 + 2\,H_2O + 4\,e^- \ .$$

The ion migration is mainly field-assisted. Not only SiO_2 emerges but also water. Water and Si again decompose into SiO_2 and other products:

$$Si + 2\,H_2O \rightarrow SiO_2 + 4\,H^+ + 4\,e^- \ .$$

Supplemental OH^- ions are supplied by the platinum cathode:

$$2\,H_2O + 2\,e^- \rightarrow H_2 \uparrow + 2\,OH^-$$
$$NO_3^- + H_2O + 2\,e^- \rightarrow NO_2^- + 2\,OH^- \ .$$

Let us now assume a constant voltage drop over the insulation layer. The local resistance of the layer is related to its local thickness. The resistance rises along with the thickness, i.e. the current carrying OH^- decreases while the layer grows. Consequently, the rate of growth is related to its thickness. Also, the field-assisted diffusion of OH^- ions depends on the layer thickness,

whereby the rate of growth is decreased in turn. Altogether, the growth of the insulation layer is self-limiting and the final layer thickness is given by the voltage U_0. As a good approximation, the thickness depends linearly on the voltage U_0 [111]. A voltage of $80\,V$ gives a layer thickness of about $50\,nm$.

The insulator quality benefits from anodic oxidation. The leakage current of a CVD oxide which is post-processed by an anodic oxidation step is reduced by two orders of magnitude compared to the leakage current of pure CVD oxide. The electric field required for breakthrough has to be larger than $10\,MV/cm$, which is a clear advancement. In this process vias having an aspect ratio of $20\,\mu m : 6\,\mu m$ are compared.

Metallisation of Vias

The metallisation of the vias starts by opening up the oxide layer which is located on top of the landing pad (the first metal layer of the metal stack). This can be done by a dry etching process with anisotropic characteristics (e.g. a CF_4 plasma).

During the etching process not only the oxide located at the via bottom will be removed. If the sidewalls are not perpendicular to the horizontal, the insulator layer will be stripped from the side walls up to a certain degree (Figure 16.20c). The critical angle γ, which must not be exceeded, is given by

$$\gamma > \arccos\left(\frac{d_{Ox,S}}{d_{Ox,S+d_K}}\right) . \tag{16.3}$$

In (16.3) $d_{Ox,S}$ denotes the thickness of the side wall oxide and d_K denotes the oxide thickness above the landing pad [102]. For typical values $d_{Ox,S} = 40\,nm$, $d_K = 200\,nm$ the critical angle is larger than $> 80.5°$. In order to circumvent

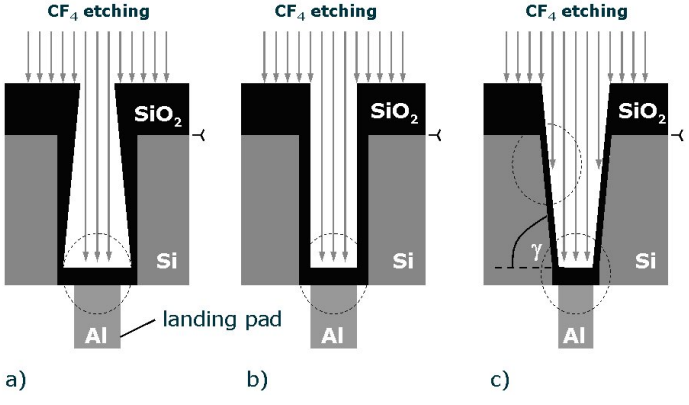


Fig. 16.20 Opening of the oxide layer above the landing pad using a dry etching process (CF_4). Three different geometrical structures are shown. Especially, in case c) the side wall oxide is affected by the etching process.

this particular problem, it is helpful to insert a modified photoresist process combined with a so-called „postbaking“ step. The postbaking step of the resist is processed after the resist has developed and taken approximately 30 min. at 100°C [102]. Then, the structured resist withstands the etching procedure and can be used as a mask for structuring the oxide. In particular, perpendicular edges within the structured resist stay stable even under etching conditions.

Copper is used as a filling material for the vias. Before the filling process starts, titanium-tungsten (Ti : W) is sputtered into the vias. On the one hand, the (Ti : W) acts as an adhesive, on the other hand it is a diffusion barrier for Cu. Ti : W is a stable barrier up to a temperature of $750°C$ and prevents the diffusion of Cu into the substrate. A diffusion of Cu into the substrate would result in deep impurities.

Prior to the electroplating process a Cu seed layer is sputtered into the via. Both, the seed layer as well as the Ti : W layer have a thickness of $20\,nm$ (Figure 16.21).

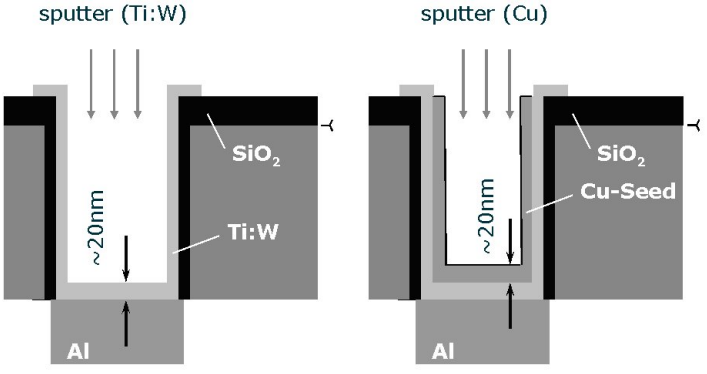
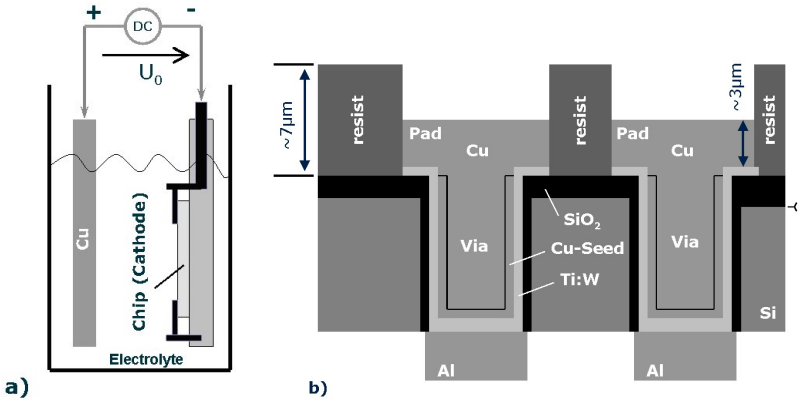


Fig. 16.21 Electroplating using Cu : deposition of adhesive, diffusion barrier and seed layer.

In the next step the via filling, as well as the metallisation of the back side contact, will be provided using a LT electroplating process (LT: low temperature). Each contact on the back side serves on the one hand as an electrical contact, on the other hand it realises the mechanical bond to the adjacent layer of the chip stack. The back side of the substrate is structured using photoresist of a thickness of $7\,\mu m$. The structure of the resist corresponds to the positions of the vias and accounts for the locations of the back side contacts (bonding pads). After the resist has been developed, an isotropic cleaning step using an oxygen plasma is inserted in order to clean the via cavities from resist contamination.

In order to fill up the via cavities, the next step includes an electroplating procedure. The reaction chamber containing a particular electrolyte for elec-



(a) Outline of a reaction chamber for Cu electroplating

(b) Cross sectional view of a sample after electroplating

Fig. 16.22

troplating is outlined in Figure 16.22a. If a voltage of about $0.35\,V$ is applied to the setup, a current results which has a density of about $56\,mA/cm^2$. After a period of 15 minutes, a Cu layer of $3\,\mu m$ thickness develops on the substrate.

The deposition rate as well as the granularity and the density of the deposited copper layer depend on the current density. In Figure 16.23 some results are shown which are obtained after an electroplating step.

In order to gain information about the electrical properties of electroplated vias, an experimental setup as shown in Figure 16.18a was analysed. The vias have a cross sectional area of $6 \times 6\,\mu m^2$ and a depth of $10\,\mu m$. Results are shown in Figure 16.24.

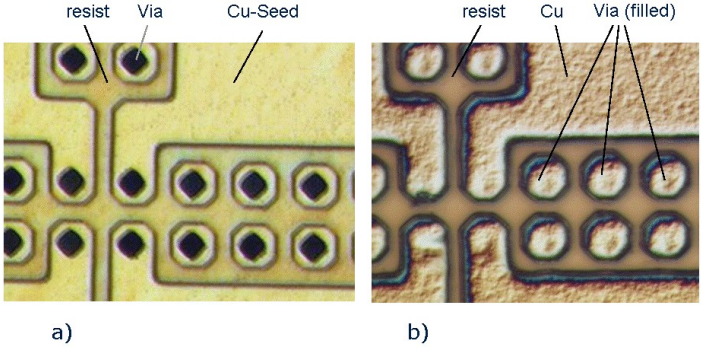
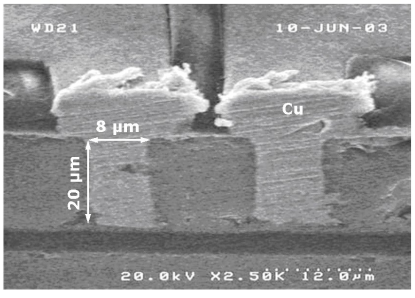
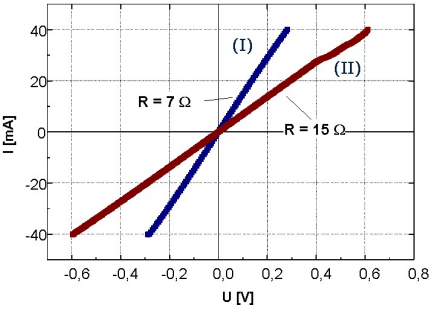


Fig. 16.23 a) Photo resist mask and vias prior to electroplating. b) After electroplating using Cu.



(a) Cross section of a sample. The homogeneity of the copper filling is clearly seen.



(b) I-V characteristics of Cu-filled vias. (I) between terminal 1 and terminal 3, (II) between terminal 1 and 2 (denomination of terminals after Figure 16.18a)

Fig. 16.24 Cross section and I-V characteristics of a sample after electroplating

The measured resistance is comparatively high in contrast to the theoretical model. Two causes come under consideration. First, the resistivity of electroplated copper is higher than the theoretical value. Secondly, junction resistance between aluminum and copper as well as resistances within the experimental setup contribute to the overall resistance. In particular, an oxide layer on top of the aluminium layer may increase the resistance considerably. A more detailed insight into this effect is given in the section about electrical characterisation of 3D contacts.

In Figure 16.25 a simple theoretical model of the via resistance is given. The model considers junction resistances at the material interfaces as well as inherent resistivities of the particular layers. The resistivity of the Cu filling is modeled by ρ, while the adhesive layer (resp. the diffusion barrier) has a resistivity modelled by ρ_B. The junctions resistance of the Ti : W barrier is given by ρ_{C1}, and ρ_{C2} gives the junction resistance between the aluminium and the diffusion barrier.

The individual resistances are connected in series. The overall resistance of the via structure can be approximated by

$$R_{via} = \rho \frac{h_{via}}{W_{via} \cdot L_{via}} + \rho_B \frac{d_B}{W_{via} \cdot L_{via}} + \rho_{C1} \frac{1}{W_{via} \cdot L_{via}} + \rho_{C2} \frac{1}{W_{via} \cdot L_{via}} \ .$$

It should be noted, that the layer thickness of the diffusion barrier is negligible compared to the width and depth of the cross sectional area. High quality layers have resistivities of $\rho_B = 50 \ldots 100\,\mu\Omega cm$ and $\rho = 1.7\,\mu\Omega cm$.

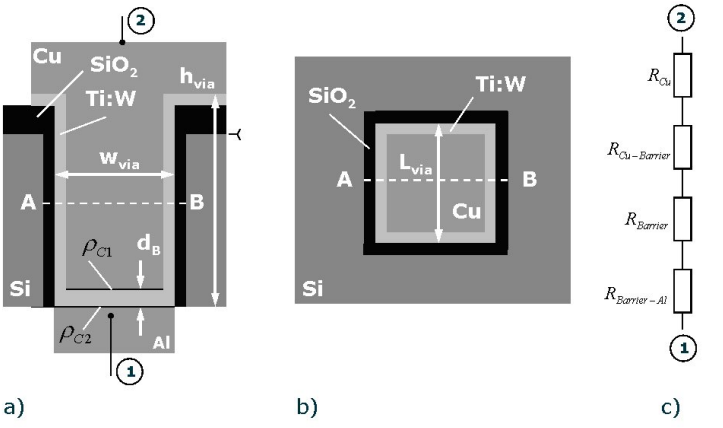


Fig. 16.25 Model of a via. Definitions of geometrical dimensions. a) Cross sectional area. b) Top view. c) Equivalent circuit diagram.

Prior to chip assembly, tin has to be deposited on the solder bonds on the back side. The tin layer will be realised using an electroplating process (Figure 16.26). The deposited tin layer has a thickness of $2\,\mu m$. The deposition is done at a voltage of $U_0 = 0.25\,V$, while the processing time was about $10\,min$. During deposition, a current density was measured at $39\,mA/cm^2$.

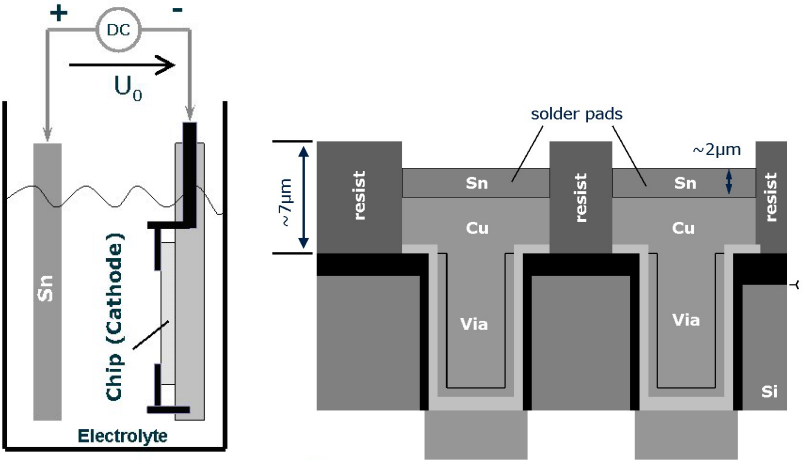


Fig. 16.26 Experimental setup of a reaction chamber for Sn electroplating (left) and cross section of a sample after electroplating (right)

16.3 Assembly

The assembly addresses the mechanical bonding as well as the electrical connection of adjacent layers. The so-called solid-liquid interdiffusion soldering process (SoLID) creates electrical and mechanical connections between single chip layers in one step. Therefore, no further processing steps are required.

The basic soldering flow is given in Figure 16.27. First, all chips are tested and diced into chips. Only the base substrate is left as a wafer. Then, the stacking process is repeated for each layer of the stack. For each iteration step, the back side contacts of the subsequent layer are adjusted to the front side contacts of the stack assembled so far. After the pads have been adjusted, the soldering process starts as soon as temperature and pressure are appropriately adjusted. After a layer has been soldered, the auxiliary carrier is removed and additional layers can be stacked. Finally, the base substrate is diced and the assembled stack is mounted in a package.

The soldering process has to meet two requirements. First, the temperature budget has to be as low as possible in order to avoid thermal degradation of CMOS during soldering. In particular, the base substrate is repeatedly exposed to the full temperature budget for each assembly iteration. Secondly,

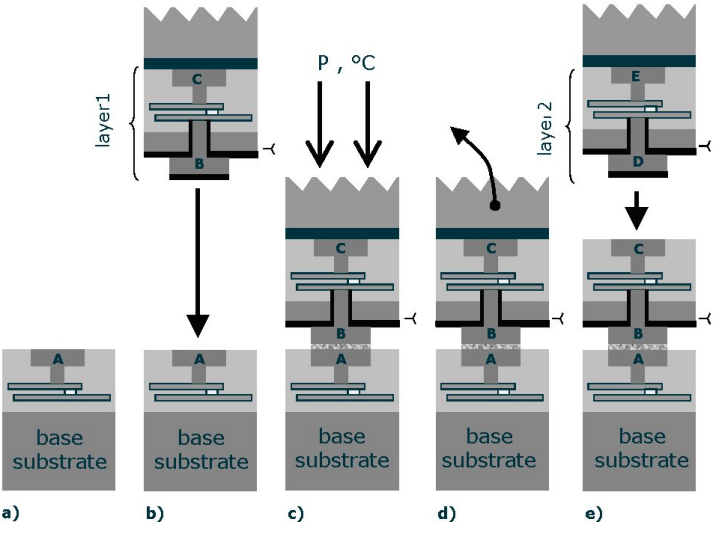


Fig. 16.27 Assembly procedure, exemplarily illustrated for layers including one front side pad and one back side pad. a) Preparation of the base substrate including a front side pad A. b) Alignment of the back side pad B (layer 1) and front side pad A. c) Pressing layer 1 against base substrate and heating up for soldering. d) Removal of auxiliary carrier. e) Alignment of pad D and pad C. Repetition of procedures c)-e), until complete stack is assembled.

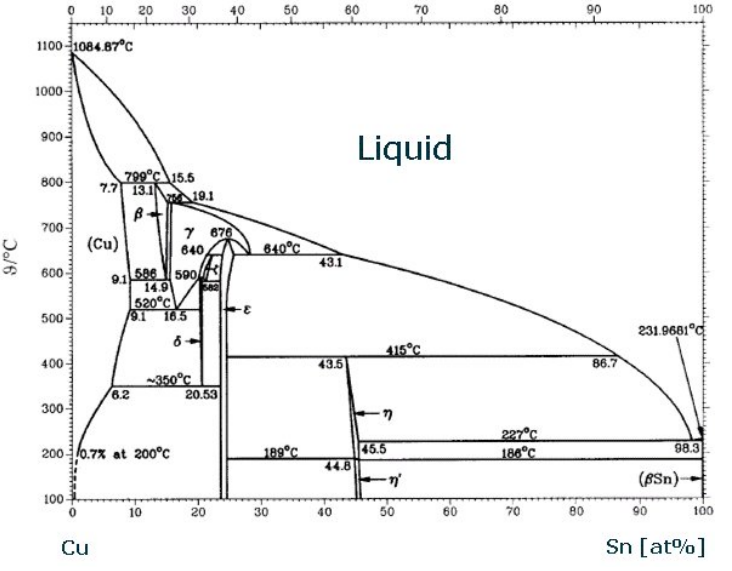


Fig. 16.28 Phase diagramm of Cu – Sn [112]

thermostability is an essential feature of the solder connections: thermostability prevents breakage of solder bonds.

Within the presented stacking process the so-called solid liquid interdiffusion (*SoLID*) soldering process is applied, which is based on isothermal solidification of an alloy composed of two different elements. One element features a lower melting point compared to the melting point of the alloy. Typical alloys are given by indium/gold or copper/tin. In the approach outlined here copper/tin is used. The phase diagram of copper/tin is given in Figure 16.28. Pure tin has a melting point of $232°C$, while the melting point of Cu : Sn is above $640°C$.

Figure 16.29a shows different phases of the alloy after melting the tin. The front side pad as well as the back side pad is realised with copper. The lower copper pad represents a bond pad of layer i (front side pad), the upper copper pad represents a bond pad of layer $i + 1$ (back side pad). In between, a tin layer is located which was deposited on the back side pad after electroplating with copper.

As soon as the layered structure is heated, the tin starts melting. Copper diffuses into the liquid tin and forms an intermediate phase, the so-called η phase. The progressive diffusion of copper into the tin changes into a thermostable ε phase. After a certain amount of copper has been diffused, the intermediate layer (alloy) is completely transformed into the ε phase. A layer thickness of $3\,\mu m$ for copper as well as a layer thickness of $2\,\mu m$ for tin has been proven to be the most advantageous for the *SoLID* process.

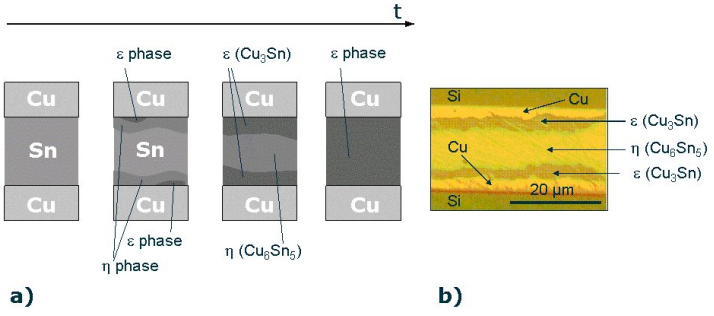


Fig. 16.29 *SoLID*: a) Formation of different phases. b) Microsection of a soldered connection.

During the soldering process, the oxidation of copper as well as tin poses a problem. In order to tackle this problem, a nitrogen atmosphere can be used in order to coat the devices during soldering [102]. In addition, prior to the soldering process copper oxide should be removed from the surface of the pads.

Though physical properties of the $Cu - Sn$ alloy are investigated for a long time, and although results are documented in various publications (e.g. [113]), for the first time ever Hübner et al. demonstrated in 2002 a process ready for production [114] which enables the assembly of two layers in a face-to-face fashion using unthinned chips and unthinned wafers, respectively. In Figure 16.30 experimental results of a two-layer assembly (unthinned substrates) are shown. These results successfully demonstrate that it is possible to realise rows of contacts with a pitch of $20\,\mu m$ between neighbouring contacts and contacts with a cross sectional dimension of less than $14\,\mu m$.

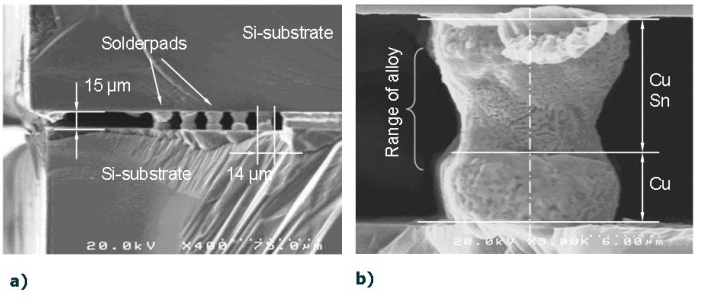


Fig. 16.30 a) Solder connection of two unthinned substrates. b) Detailed view of a solder contact.

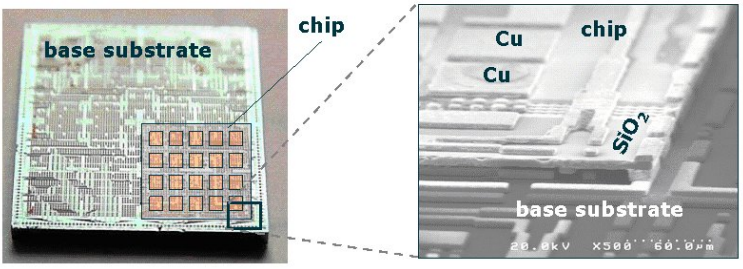


Fig. 16.31 Two-layer stack. a) chip photograph. b) detailed view.

In Figure 16.31 two individual substrates are soldered. In contrast to Figure 16.30 the top layer was thinned. After soldering the auxiliary carrier was removed from the top chip and only the thinned layer is visible. The removal of the carrier was done by immersing the sample into a 1-methyl-2-pyrrolidone solution at a temperature of $150°C$. A time frame of $30 - 60$ minutes is required to detach a carrier with an area of $5\,mm^2$. After the carrier has been detached, a cleaning step is inserted in order to purge the surface of the chip from residual adhesive. This was done by exposing the surface to a O_2 plasma. The sample in Figure 16.31 shows the surface of the chips, the metallisation and several measurement pads.

So far we were able to realise a mechanically stable six-layer stack (Figure 16.32).

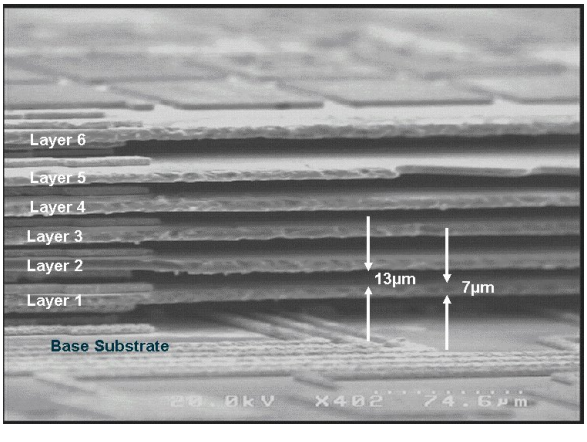


Fig. 16.32 REM image of a six-layer stack.

16.4 Electrical Properties of 3D Interconnections

This chapter presents experimental results with focus on DC properties of 3D connections. In particular, issues about

- resistance of solder connection,
- resistance of vias,
- resistance of copper wires

were examined. In Figure 16.33 an experimental setup is given which allows the determination of the restistance of solder connections. The solder connections were designed with an cross sectional area of $12 \times 12\,\mu m^2$. The total resistance of the setup is given by the resistance of the solder connection and the resistance of the supply lines. In order to compensate for the resistance of the supply lines, the total resistance is measured for different lengths of the supply lines which allows an estimation of the supply resistance. After compensation it turns out that electroplated copper has a resistivity of $\rho_{Cu,Measurement} \approx 3\,\mu\Omega cm$. The solder contact has a resistance of $R_{Contact} \leq 0.1\,\Omega$. From the literature $\rho_{Cu} \approx 1.7\,\mu\Omega cm$, $\rho_{Cu3Sn} \approx 8.9\,\mu\Omega cm$ and $\rho_{Cu6Sn5} \approx 17.5\,\mu\Omega cm$ were expected.

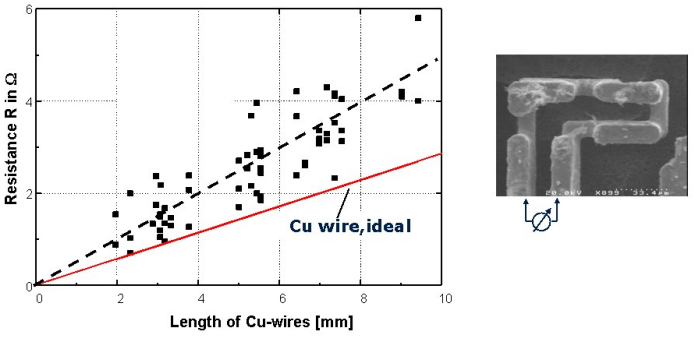


Fig. 16.33 Resistance of electroplated copper wires. The wires were bypassed by a bridge. The dimensions of the bridge are $12\,\mu m \times 12\,\mu m$.

In Figure 16.34 a setup is shown, which permits the determination of the via resistance. The setup is made up of two vias, one bridge, and two pads for measurement. The given structure is realised on a single substrate, i.e. without soldering. After compensation for the measurement setup, a resistance of $R_{Via} < 0.7\Omega$ for a $6 \times 6\,\mu m^2$ via was obtained. Compared to results presented in Figure 16.23, the via resistance is much lower. This can be explained by a higher contact resistance between copper metallisation and the aluminium electrode.

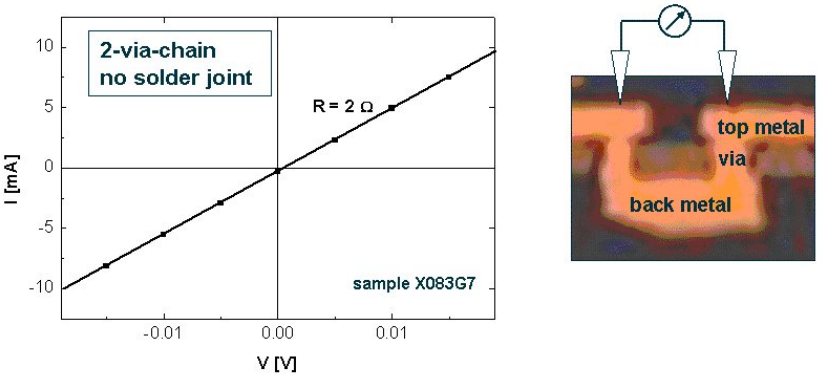


Fig. 16.34 Resistance of a two-via-chain. The structure is realised on a single substrate with copper wires on the front side and a copper bridge on the back side. There was no soldering involved.

A setup, which includes an additional solder connection acting as a bridge, is shown in Figure 16.35. The supply lines of length $2.8\,mm$ were realised using copper. From results presented so far, the resistance of the supply line can be estimated to $1.18\,\Omega$. From calculations, the overall resistance, including via and solder connections gives $R_{Via,L"ot} < 0.8\,\Omega$. This result perfectly matches other results obtained for the resistance of solder connections as well as vias.

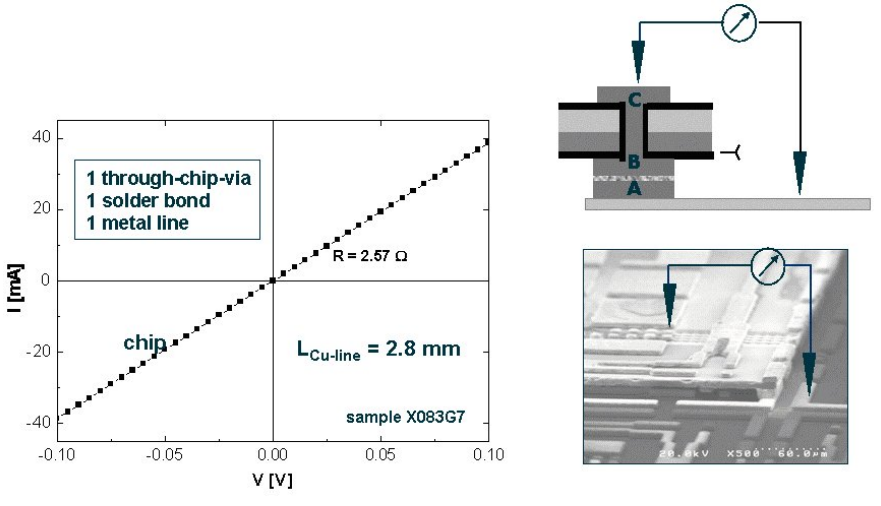


Fig. 16.35 Resistance of a through-via. The resistance of the supply line ($2.8\,mm$ length) adds to the overall resistance.

16.5 Summary

The proposed 3D stacking technology is divided into four key processes:

(i) nondestructive thinning of substrate,
(ii) precise alignment of front side pads to back side pads,
(iii) preparation of back side vias (etching, insulation, metallisation),
(iv) assembly (soldering by means of isothermal solidification).

All these processes were developed in a co-ordinated way and permit the realisation of multi layer architectures in a mechanically stable way. As a prerequisite, all processing steps are low-temperature steps in order to avoid device degradation:

- the process for bonding and detaching the auxiliary carrier reaches temperatures in the range of $250°C - 270°C$ for a time interval of $4h$,
- the oxidation requires temperatures of less than $200°C$ for less than $3h$,
- the solder process requires temperatures below $300°C$ and only for a short time.

There is no processing step within the backend processing chain which requires temperatures higher than $300°C$. Wafers can be precisely thinned up to a residual thickness of $10\,\mu m$. The alignment of masks as well as of pads during assembly can be realised at better than $1\,\mu m$ using optical equipment. The cross sectional area of soldering pads is less than $6 \times 6\,\mu m$. The pitch of neighbouring pads is less than $15\,\mu m$.

To sum up, on an area of $1\,mm^2$ more than 4.400 3D pads can be placed. A chip having an area of $50\,mm^2$ potentially holds more than 250.000 vertically orientated contacts. Needless to say, due to this high number of vertical interconnections, additional questions concerning 3D stacking arise. Efficient test procedures for a 3D stack have to be developed. The exploitation of redundancy for yield enhancement after testing is a fundamental challenge for further investigation. Rises in temperature due to power consumption as well as power dissipation are further major challenges for 3D stacks in general. Finally, novel design tools (semi-automatic, fully automatic, for partitioning, simulation, place and route) have to be provided in order to facilitate the design process of 3D systems.

The proposed stacking technology provides the fundamentals for the implementation of small-sized vision cubes for mobile applications. An example of a vision cube is presented in the next chapter.

17

Architecture of First Generation Vision Cube

Arne Heittmann and Ulrich Ramacher

We now want to determine the number of different modules which are required to build an artificial vision system which achieves the resolution of the retina, possesses 8 orientations per resolution plane and 5 resolution planes and recognises distant-invariantly up to 1000 objects each featuring 10 zones. The neural architecture of this vision system is described in chapters 6-9.

17.1 Imager Chip

In order to realise the desired resolution for layer A (Figure 9.5) the imager circuit has to include an array of 1024×1024 pixels. As explained in section 12.3 a pixel cell with a minimum-sized pixel cell fires with a rate of $3\,kHz$ when saturated, though $300Hz$ would do. A rate divider (factor of 10) consisting of a few flip-flops is added therefore. If we assume an area of $5 \times 5\,\mu m^2$ for the pixel cell alone, the overall area, including the neuron and the rate divider, can be estimated at $100\,\mu m^2$ (for 65 nm CMOS technology). Then the area of the whole array is estimated at $100\,mm^2$ which covers the effort for layer A only. The organisation of the sensor array, as well as the basic circuit of the pixel cell, is shown in Figure 17.1. The circuit for the pulse formation is taken from Figure 12.34. A global adjustment of the pulse frequency to the luminance conditions is possible by choosing a reference voltage U_{ref}.

Because of a low yield, the residual layers of the resolution pyramid (Figure 9.5) can not yet be integrated on this chip. But a sufficiently large area exists on the backside of the imager which can be used to attach more than 1 million vertical connections. These connections can be used for the transmission of pixel information to the next layers. This chip which is adjacent to layer A receives pulses directly from layer A and derives thereof the pulsed pixel layers of the resolution scales B-E. Additionally, this chip contains the AER encoders which exclusively link the image layers A-E to the feature detection layers. Since this chip is mainly used to encode pulses into an AER representation, this chip is called AER-chip. 300 million pulses per second will

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 299–323, 2010.
© Springer-Verlag Berlin Heidelberg 2010

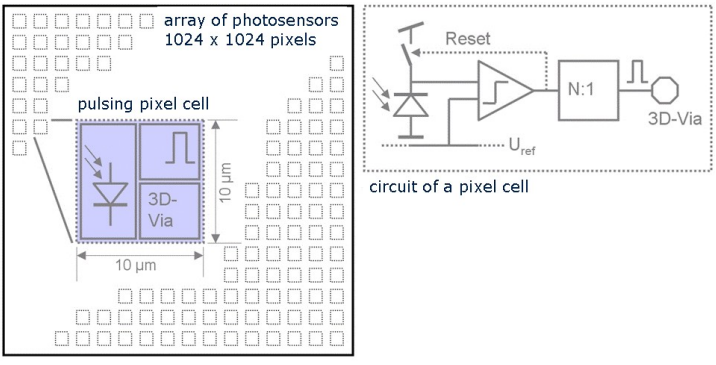


Fig. 17.1 Block diagram of a photo sensor chip with 10^6 3D vias

be transmitted via 1 million vertical vias to the AER chip under saturation conditions. Note that the pulse transmission from the imager to the AER-chip needs no additional control nor a dedicated communication protocol.

17.2 AER Chip

The stream of pulses which is transmitted from the imager chip to the AER-chip is divided into two separate streams. The first stream directly addresses the AER unit, the second stream is linked first to circuits, which are used for the formation of the resolution pyramid, and then directed to the AER unit.

The AER unit replaces an incoming pulse with a dedicated code word. This code word uniquely specifies the location on the retina where the pulse has been generated. In contrast to the implementation described in chapter 14 the coding is simplified with respect to compression: the code word is not compressed and no time stamp is added (the reason being, of course, the increased connection density provided by 3D stack technology). In the following, we use the term „pulse event“, for a pulse, as well as for a code word generated by an AER unit.

As explained later, the A-layer is divided into 4096 so-called segments with 256 pixels each. Segments overlap around the boundary area by 3 pixels, since also pixels located at the boundary are the origin of wavelets with a receptive field of 7×7 pixels. Hence each segment receives input from $(16 + 6)^2 = 484$ pixels. Each segment has its own AER unit and receives pulses from the imager via 256 vertical vias. Each pulse neuron of a pixel cell is connected to a so-called signalling cell. The signalling cell in turn is connected to an arbitration circuit which schedules incoming events triggered by the signalling cell (Figure 17.2 left: the pulse signal of the pixel cell is connected to the signalling cell via a vertical via). The arbitration circuit is composed of two priority encoders, which encode the X-coordinates as well as the Y-coordinates

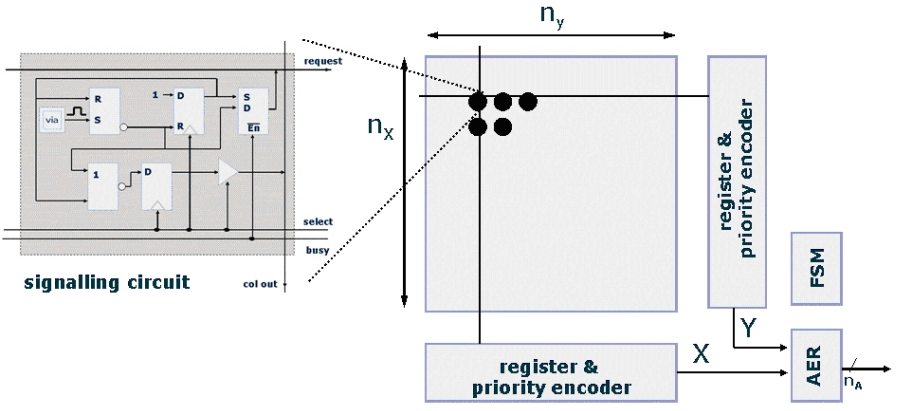


Fig. 17.2 Circuit for encoding of pulse events. (Vias as well as signalling circuits are illustrated using black circles)

of incoming pulse events. The number of bits required to uniquely encode the coordinates is given by $n_A = |\log(n_x)| + |\log(n_y)| = 10$.

The AER-circuit can be reused unaltered from the design of the spot detector (chapter 14). The area requirements of a signalling cell can be estimated to 18 logic gates. Thus the overall area for the array is given by $n_x \times n_y \times 18$ logic gates. All in all each priority encoder covers an estimated area of $(n_x + n_y) \times 22$ logic gates.

Vias and signalling cells, respectively, are sampled with a sampling rate of $484 \times 300\,Hz = 145.2\,kHz$. The clock frequency required for the transport of the pulses into the feature cascade depends on the number of available vertical vias, as well as on the number of bits which have to be transmitted. If we add n_S bits (for segment identification) to each code word, and if n_V vias are available for data transmission, the minimum clock frequency for the propagation of one pulse event is given by $145.2\,kHz \times (n_A + n_S)/n_V$.

The residual layers of the resolution pyramid include so-called averaging neurons as well as AER units for each layer B-E. In this regard we refer to neurons in layers B-E as „pixels“. Similar to layer A, the layers B-E of the resolution pyramid are divided into segments which consist of 256 pixels. The entire resolution pyramid contains 5456 segments. If we utilise 10 vias for each segment, the required clock frequency for data transmission is given by 435.6 kHz. The AER-chip and the adjacent chips for the feature cascade are connected by $5456 \times 10 = 54560$ vias.

Next we describe the effort required for the realisation of the AER-chip. The floorplan of the AER chip is sketched in Figure 17.3. The AER chip contains $64 \times 64 = 4096$ segments. Each segment is linked to $5456 \times 256/64^2 = 341$ pixels, which are located in layers A-E.

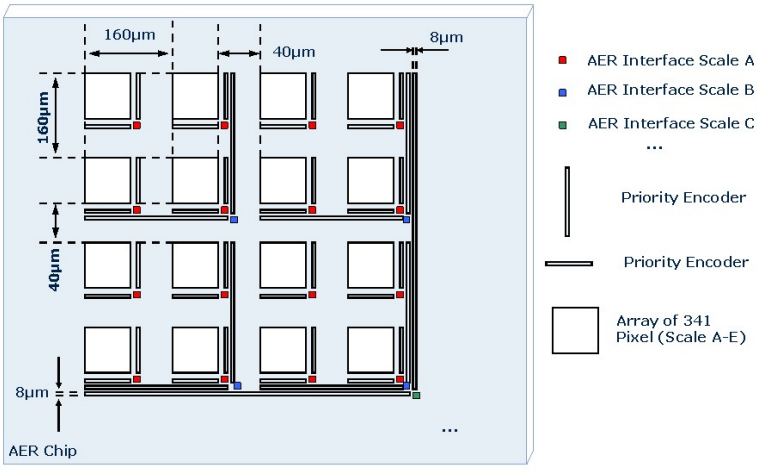


Fig. 17.3 Floorplan details of the AER-chip which realises the resolution pyramid. Segments are placed on a fixed grid in order to simplify wiring between AER chip and CMOS imager.

As mentioned, each pixel in the A-layer comprises a signalling circuit (Figure 17.2). In addition, each pixel in the layers A-D is connected to an averaging circuit, which is used for the formation of pixels which are located in the respective resolution layer B-E. This averaging circuit receives input from four neighbouring pixels.

We now consider a particular section of layer A containing 16×16 pixels. 16×16 pixels of layer A are used to derive 8×8 pixel for layer B, 4×4 pixels for layer C, 2×2 pixels for layer D and 1 pixel for layer E by repetitive local averaging of pulse rates. The wiring architecture of such a segment is illustrated in Figure 17.4.

The individual pixels of the AER chip were placed on a regular grid in order to obtain as compact a layout as possible for the hardware realisation. A signalling circuit containing 18 logic gates (or gate equivalents) is attached to each pixel in the A-layer. The remaining pixel cells in the layers B-E contain an additional averaging circuit (Figure 17.5). If we estimate 60 logic gates for the averaging circuit, the whole segment of 16×16 pixels contains roughly $341 \times 18 + 85 \times 60 = 11238$ logic gates. The required area is estimated to be $7866\,\mu m^2$. If we consider a pixel distribution according to Figure 17.4, the logic gates of the segment fit into an area of $118 \times 66.5\,\mu m^2$. The remaining area of $6534\,\mu m^2$ can be optimised with respect to the wiring of all particular signalling cells and with respect to the connection of the priority encoders of all resolution layers.

Particular priority encoders which encode pulse signals are integrated for each resolution layer. The pulse signals either originate from neurons which

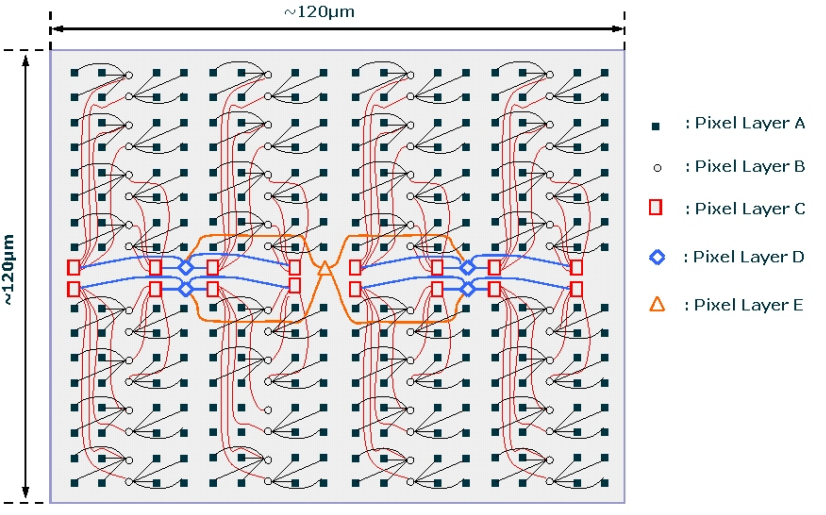


Fig. 17.4 Wiring of a cell with 16×16 pixels. The cell is used for the formation of resolution layers A-E. The subcircuits are placed on a 20×18 grid. The overall number of pixels which are contained in this cell is $256+64+16+4+1 = 341$. With respect to layout constraints, the whole circuit had to be placed within an area of $120 \times 120\,\mu m^2$.

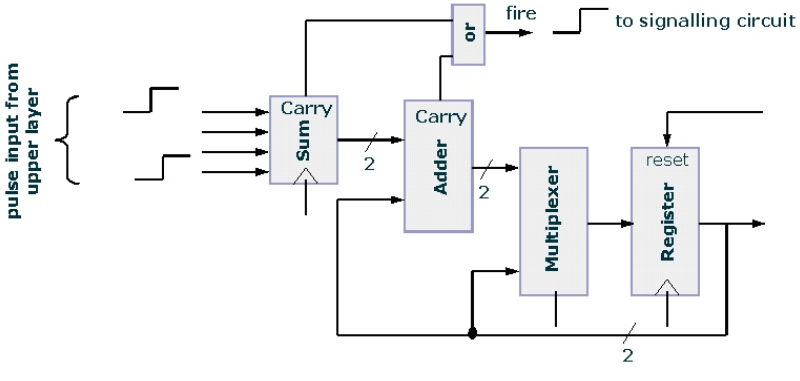


Fig. 17.5 Outline of a 4-input circuit which is used for averaging of pulse rates. For each cycle, the circuit *Sum* counts the concurrent number of rising edges which are related to the four input signals. This number is continuously added to the register. If the accumulated number of edges is larger than (or equal to) 4, a pulse is generated.

are located within the respective segment or originate from neurons which are located in a neighbouring segment within the overlapping area. The encoded pulse events are transported via an AER interface to the adjacent layers of the feature detectors. Subsequently we assume five resolution layers for the resolution pyramid. Each priority encoder is designed with a fixed width and height respectively, of $8\,\mu m$, whereby an area of $120 \times 120\,\mu m^2$ remains for the implementation of the pixel array. Each priority encoder receives $16 + 2 \times 3$ input signals which originate from the pixel array. Thus, 968 logic gates are needed in each segment for the implementation of the priority encoders. But the area for the logic gates is not so important, since the area of the priority encoders is determined by the wiring. The specified width of $8\,\mu m$ is given by the area requirements regarding the wiring of the priority encoder.

17.3 Size of the NPU Array and Memory Requirements

In order to create detector profiles with the required flexibility, the hardware components for the feature detectors must provide adequate configuration capabilities. Then, arbitrary network architectures can be realised by programming and re-programming of the network configuration at runtime.

The very simplest architecture model of the feature cascade would contain two NPU elements: NPU1 processes the neurons N_{xa} of the detector architecture and NPU2 processes the neurons N_{xb}, respectively ($x = 3, 4, 5, 6, 7$). The basic architecture of a NPU element was already outlined in section 15.2. An important parameter of the architecture is the clock frequency which will be determined subsequently.

We consider a single wavelet orientation. The detector layer receives a data stream of $1.26\,GB/s$ peak from the AER chip. Each pulse is encoded by 3 bytes. As already mentioned, each pulse event which originates from the AER chip triggers off 20 particular detectors in the feature detection layer. The membrane potentials of neurons $N_{3a,b}$, $N_{5a,b}$, N_{5b} and $N_{6a,b}$ have to be updated for each detector triggered by an incoming pulse event. The pulse activities of neurons $N_{3a,b} - N_{7a,b}$ have to be derived from the state of the membrane potentials. Both pulse activity as well as membrane potentials are calculated using the NPU architecture according to section 15.2. It is assumed that all detectors share a particular NPU pair. Hence, all states, as well as all membrane potentials, have to be stored in a separate memory. State information has to be read out from the memory and will have to be modified by the NPU elements for any incoming pulse event. Finally, the results are stored back in the memory. If particular $N_{7a,b}$ neurons generate pulse events, the activity will be AER-encoded and transferred to the next layer via an AER interface on the output side.

The complete update of the membrane potentials for a particular detector requires five subtasks which can be executed within five clock cycles:

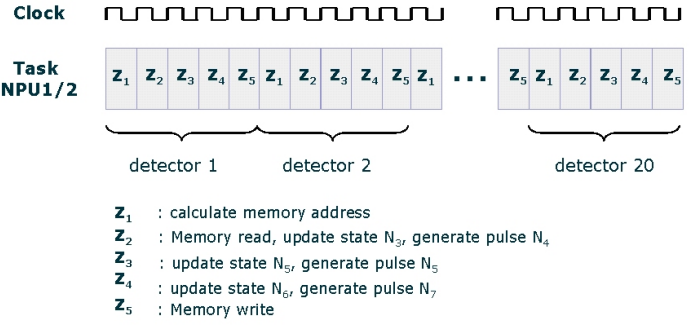


Fig. 17.6 Individual tasks for the formation of the detector response by the NPUs

Z_1 calculation of memory address which is used for storing states of the actual detector,
Z_2 readout of state information from the memory, update of the state information for neuron $N_{3a,b}$, formation of a pulse for neuron $N_{4a,b}$, if needed,
Z_3 update of state information for neuron $N_{5a,b}$, and formation of a pulse for neuron $N_{5a,b}$, if needed,
Z_4 update of state information for neuron $N_{6a,b}$, and formation of a pulse for neuron $N_{7a,b}$, if needed,
Z_5 rewriting of state information into memory.

After 100 clock cycles the set of 20 detectors is updated (Figure 17.6).

The resolution pyramid contains 1 396 736 pixels, i.e. under saturation conditions the detector receives 419 020 800 pulses/s. With 100 clock cycles per incoming pulse event from the AER chip, the clock frequency of the NPU pair has to be larger than $\approx 42\,GHz$. This is not realisable, neither from the perspective of power dissipation, nor from the perspective of system timing. Measures have to be taken in order to lower the clock frequency considerably.

The first step in lowering the clock frequency is to double the number of NPUs. Then, half of the detector responses will be determined by a particular NPU pair in parallel. If the execution time is kept constant, the clock frequency can be reduced by a factor of two. Since we assume that the memory interface can handle only one access (read or write) per clock cycle, a skew of one cycle between the parallelized tasks emerges (Figure 17.7).

4 out of 5 clock cycles are used for the memory access. The optimum memory interface utilization (one read-write access in each clock cycle) is accomplished if three NPU pairs are integrated. However, this approach produces an additional empty cycle (NOP) which has to be inserted for each NPU pair. The utilization of the NPUs decreases, which is not desirable. Any further parallelization beyond a factor of three necessitates additional empty cycles, which degrades the NPU utilization more and more. Hence we like to

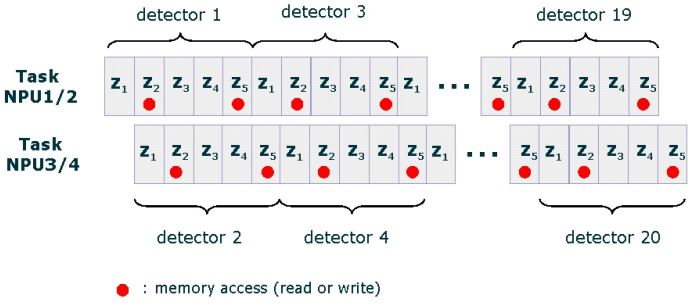


Fig. 17.7 Formation of detector response by interleaved operations

keep to just two NPU pairs. In the following, a set of two NPU pairs is called a „NPU-Quad“.

Now, only $1 + 10 \times 5 = 51$ clock cycles are required for the processing of 20 detectors, meaning the clock frequency of the NPU quad has to be larger than $\approx 21\,GHz$ under saturation conditions. Again, this clock frequency is still far too high.

Instead of assigning the full sensor array of 1024×1024 pixel to a single NPU quad, it is reasonable to divide the array into smaller segments which in turn are attached to several NPU quads. For instance, if we divide the array into two segments, two AER interfaces are required which transfer pulse events at about half the rate. Due to a 3 pixel overlap of the receptive fields (7×7 pixels), each subarray contains 515×1024 pixels. This approach is illustrated in Figure 17.8. By repeating this „segment-and-reduce“ approach, a multi-segmented architecture emerges (cf. Figure 17.9 for some examples).

Table 17.1 illustrates the required clock frequency for different segmentations. The calculations are based on the effort which is required for the realisation of the A layer. From the viewpoint of system timing, architectures containing more than 64 segments are feasible.

If the number of segments is increased, the area demands for additional logic have to be considered. Each segment has its own NPU element, as well as logic for its own AER unit. Next we will estimate the memory requirements, as well as the number of logic gates for different architectures with

segments	1	64	256	1024	4096
NPUs (1 wavelet)	4	256	1 024	4 096	16 384
number of pixels[kPixel] (1 Segment)	1 048.58	17.96	4.90	1.44	0.48
event rate per segment [MEvents/s]	314.57	5.39	1.47	0.43	0.15
f_{System} [MHz]	16 043.21	274.73	74.97	22.09	7.41

Table 17.1 Dependence of clock frequency on the segmentation

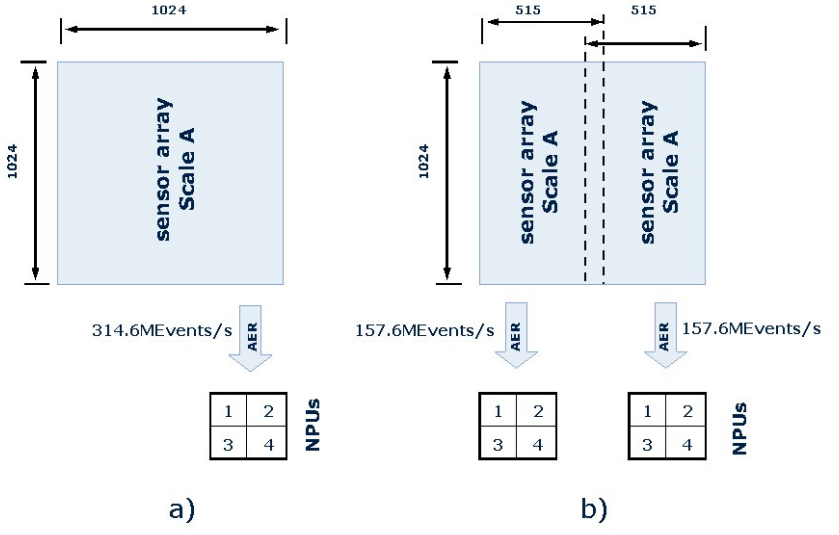


Fig. 17.8 Reduction of clock frequency by exploiting segmentation of the imager.

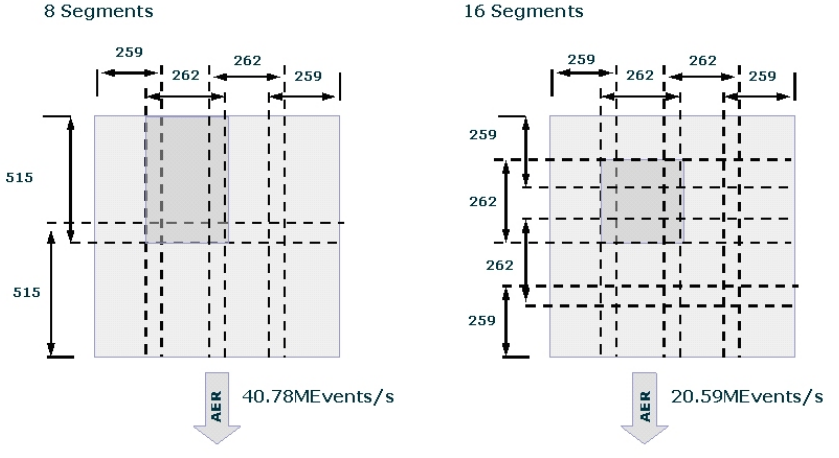


Fig. 17.9 8/16 segments architecture

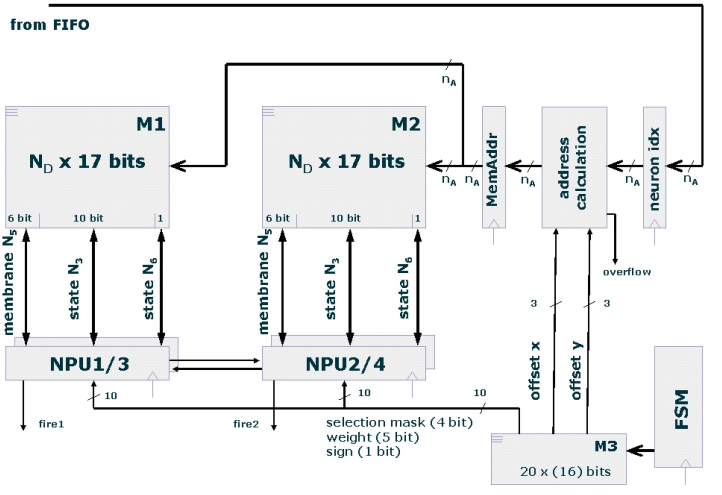


Fig. 17.10 Architecture of 4 NPUs processing a single orientation

respect to segmentation. Using these numbers, the overall area demands can be estimated.

The NPU units successively update the state information (i.e. the membrane potential) of interior neurons, which belong to particular detectors, on each incoming pulse event. The information about the membrane potentials is stored in the memory. A respective architecture is shown in Figure 17.10 which contains memory as well as NPU processing elements. The memory blocks M1 and M2 store the entire state information of the interior neurons for the set of detectors which belong to a particular segment (i.e. for one wavelet orientation and one resolution scale). The processing elements NPU1 and NPU3 update the membrane potential of neurons N_{xa}, and the processing elements NPU2 and NPU4 update the state information of neurons N_{xb} ($x = 3, \ldots, 7$). The access to the memory from NPU1 and NPU3 is separated by one clock cycle. The same scheme applies to the NPU pair NPU2-NPU4. Finally, the connection between NPU1/3 and NPU2/4 facilitates the state update of neurons $N_{6a,b}$.

The NPU design given in section 15.2 can be reused almost unchanged. Compared to Figure 15.5, the NPU design is complemented by additional „A-before-B“detectors. Here, 11 such detectors (Figure 17.11) are integrated, which enables the parallel processing of 22 gradient detectors by a single NPU pair. 10 out of 11 A-before-B detectors are utilized for the processing of the neurons N_3, the remaining one processes neuron N_6. With the help of the results presented in Figure 15.5, the complexity of a single NPU can be assumed to comprise 580 logic gates.

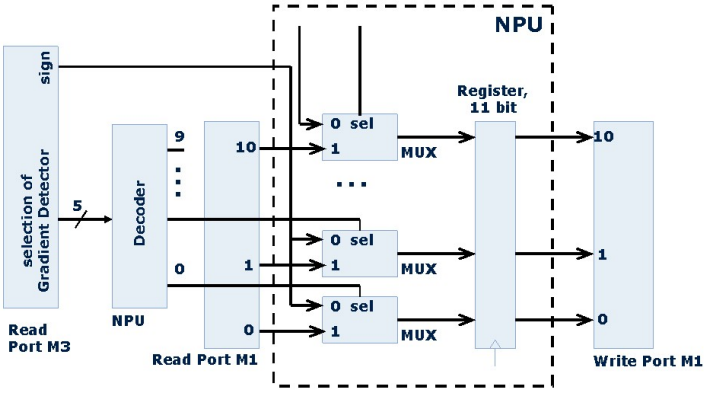


Fig. 17.11 Extension of a NPU by 11 „A-before-B“ detectors

The required complexity for the memories M1 and M2 can be estimated as follows. As in the case of the NPU test chip (section 15.2), the membrane potential of the neurons $N_{5a,b}$ is represented by 6 bit data words. It is sufficient to encode the precharge state only for the neurons $N_{3a,b}$ (and accordingly $N_{6a,b}$). Only two states have to be represented: (i) „precharged“, and (ii) „not precharged“. The distinction of both states is done with the reservation of a single bit for the state representation, i.e. one bit of memory is needed per gradient detector. Since 22 gradient detectors are integrated in the detector architecture, the overall amount of memory is given by 2×17 bits per detector. With a quantity of N_D detectors per segment, in general a total capacity of $N_D \times 34$ bit is required for M1 and M2.

Finally, we describe how to address M1 and M2. Addressing depends on the location of the respective detector within the segment. Similar to pixels, the detectors are grouped in the detector layer as a 2D array. A given detector is characterised by its coordinates (X, Y) relative to the origin of the array. We relate the coordinates of pixels to the coordinates of detectors. If an incoming pulse has given coordinates (X_p, Y_p) all coordinates of assigned detectors can be calculated by adding an $X-$offset as well as an $Y-$offset to X_p, Y_p, as we will explain now.

Since for an incoming pulse the state information of 20 detectors has to be updated, the 20 detectors are given by the distribution of the sampling points of the respective wavelet. Each sampling point can be specified by an individual coordinate relative to the origin of the receptive field. Then the $X-$offset, as well as the $Y-$offsets to (X_p, Y_p) are given by the coordinates of the sampling points. If a receptive field of 7×7 pixels is assumed, the $X-$offsets and the $Y-$offsets range from $-3 \ldots 3$ and can be represented by 3 bits. For each sampling point the memory M3 (Figure 17.10) contains the appropriate X/Y

coordinates. These values are processed by the module *address calculation*. The input to this module derives from the coordinates (X_p, Y_p) of the incoming pulse, as well as the offset propagated from M3. The output is the absolute memory address used to access the state information stored in M1/M2 and which must be updated proactively. An integer weight is provided by M3 in addition to the offset which specifies the so-called multiplicity of the sampling point. In order to sufficiently represent the multiplicity of sampling points, 5-bit data words are required. Equally, with 20 sampling points, 5 bits are required to specify the „A-before-B" detector whose state has to be updated. Because for each sampling point, the overall amount of memory is given by 16 bit, M3 must have a capacity of 320 bit. The block *address calculation* requires approximately $19 \times n_A$ logic gates.

17.4 Chip for Feature Detection

Detectors with differently orientated wavelets can be implemented by replicating the proposed architecture. The specific wavelet profile will be defined at runtime by initialization of the memory M3. On the input side pulse events are transported from the FIFOs associated to a particular segment to the adjacent detectors. In Figure 17.12 there is an illustration of an architecture suitable for the processing of 8 different orientations. Since each segment contains 8 detector circuits, the effort required to connect the detector layer and the feature recognition layer is 8 times higher than for the connection of the sensor layer and the detector layer. On the other hand, the bandwidth requirements are relaxed since only one subset of local detectors will respond to the given input. If we assume that the detectors gather nearly complete and non redundant information from the pixel layer, the required output bandwidth roughly matches the input bandwidth. Consequently, it is sufficient to provide a single AER interface on the output side. Before pulse events are transferred via the AER interface, the pulses will be prepared appropriately by the multiplexer (Figure 17.12).

Table 17.2 reports the effort required for different segmentations of the A-layer.

The area estimations are based on the assumption that 1 MByte of memory occupies an area of $3\,mm^2$ while one logic gate occupies an area of $0.7\,\mu m^2$ (both 40nm CMOS). Additionally, each 3D via accounts for $5 \times 5\,\mu m^2$ active area (for the landing pad) as well as for $10 \times 10\,\mu m^2$ on the surface (for the solder pad), see chapter 16.

If more layers of the resolution pyramid are added, additional NPU units as well as AER channels and FIFOs on the input side have to be integrated. Area estimations for the whole feature cascade are given in Table 17.3.

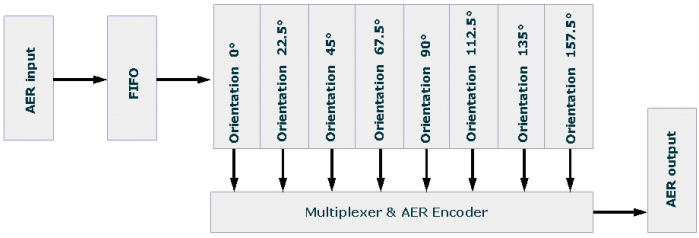


Fig. 17.12 Architecture of one segment. Each segment provides detectors for 8 different orientations. The output signals of individual detectors are transferred to the feature recogniser via the AER-output interface using multiplexed operations.

segments	1	64	256	1024	4096
NPUs	32	2 048	8 192	32 768	131 072
NPUs gates [MGates]	0.02	1.34	5.30	20.87	82.25
AER [MGates]	18.92	19.25	19.66	20.59	22.84
FIFO [MByte]	2.50	2.19	2.09	2.12	2.36
memory M1,2,3 [MByte]	34.00	34.02	34.08	34.31	35.25
3D vias (Data)	20	1 024	3 584	12 288	40 960
area NPUs [mm^2] (40 nm)	0.02	0.94	3.71	14.61	57.57
area Memory [mm^2] (40 nm)	109.50	108.63	108.51	109.28	112.84
area 3D vias [mm^2] (landing pads)	0.0005	0.0256	0.0896	0.3072	1.0240
area 3D vias [mm^2] (solder pads)	0.0020	0.1024	0.3584	1.2288	4.0960

Table 17.2 System requirements for different segmentations for scale A only. The imager has 1024×1024 pixels. Feature detectors for 8 different orientations are covered by the architecture.

segments scale A	segments in scales B,C,D,E	NPUs	area [mm^2] NPUs	memory [MByte]	area [mm^2] Memory	3D vias (data)	area [mm^2] detector	area [mm^2] AER chip
1	1,1,1,1	160	0.08	48.62	145.86	100	145.94	32.28
64	16,4,1,1	2 752	1.26	48.66	145.98	1 376	147.28	32.84
256	64,16,4,1	10 912	4.94	48.18	144.55	4 774	149.60	33.51
1024	256,64,16,4	43 648	19.46	48.52	145.57	16 368	165.44	35.02
4096	1024,256,64,16	174 592	76.69	50.10	150.31	54 560	228.36	38.69

Table 17.3 System requirements for 8 segmentations and 5 resolution scales A-E. The number of 3D vias relates to the number of connections between the imager chip and the detector chip.

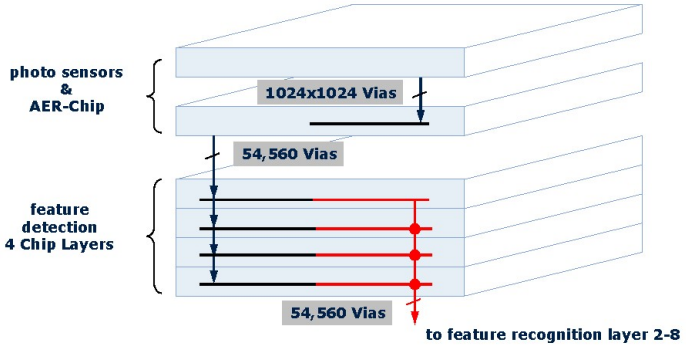


Fig. 17.13 Block diagram of the first layers for the vision cube

3 chip types and 6 chips are required for the assembly of the feature cascade in the case of 4096 segments (cf. Figure 17.13): 1. one imager chip (about $100mm^2$), 2. one AER chip (about $64\,mm^2$), and 4 NPU chips (each about $64\,mm^2$).

The AER chip is divided into several domains. An area of $38.7\,mm^2$ is used for the implementation of digital logic for the AER units. The overall area for landing pads (which are connected to solder pads in turn) is less than $2.7\,mm^2$. The residual area of $22.6\,mm^2$ serves for additional circuits which interface the feature cascade with computers and the like (i.e. interface logic, logic for configuration and control, memory).

17.5 Chip for the Feature Recogniser

The next layers of the processing chain include the positioning of zones and the binding of features to objects on each resolution scale. In the following we will discuss the effort needed to produce the hardware of a first-generation architecture which is capable of recognising 1000 different objects. It is assumed that an object is described by 10 zones (for review of the network architecture of a particular zone cf. section 7.2, section 7.3 and section 7.8). Let us remember that each zone is comprised of 20×24 neurons.

The circuit of the feature recogniser performs different tasks on incoming pulses which are generated in the feature detection layer. Individual pulses must be associated with individual zones, and within a given zone the following sub-tasks are carried out:

- O1: averaging of the pulse rate for the spatial-temporal input pattern (using neuron N_a, cf. Figure 17.14 and 7.6),
- O2: difference formation between detector rate and average rate,
- O3: synchronisation of pulses and recognition of features using the synchronisation detector.

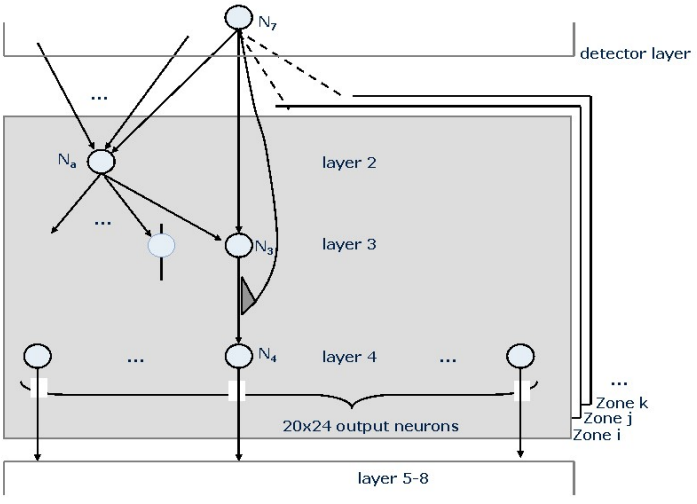


Fig. 17.14 Zone architecture

With regard to O1, a particular neuron N_7 of the feature detection layer may be connected to several zones. Each pulse of neuron N_7 will be sent towards the averaging neuron N_a as well as towards an adjacent neuron N_3 in layer 3. If an incoming pulse causes neuron N_a to fire, all neurons N_3 of a zone will be precharged via dynamic synapses (Figure 7.6).

We prefer a digital circuit for the implementation of operations O1 and O2 in order to provide the required flexibility (as it has been the case for the feature detector). An analogue circuit is used for operation O3 since it definitely provides a more economical realisation (cf. chapter 14).

First of all, we will have a closer look at the operations O1 as well as O2 and investigate the architectural choices. We will start with the design of a circuit representing a particular zone. This circuit should exhibit full flexibility: the resolution layer as well as orientation and location of the assigned feature neurons can be arbitrarily specified. Only the number of input signals is limited to $20 \times 24 = 480$ neurons. The particular assignment of feature neurons to a zone is done at runtime by individual configuration. The full set of zones is realised by repetitive implementation of the proposed prototype architecture.

An AER interface is located on the input side, which receives pulse events from the detection layer (N_7neurons, Figure 17.15). Each pulse event uniquely encodes the origin of a pulse in terms of the resolution layer and its associated orientation, as well as the X-Y location, by means of an index. The index is presented as $20 + 3 + 3 = 26$ bits. The orientation (1 out of 8) as well as the resolution layer (1 out of 5) are each encoded by 3 bits. The remaining 20 bits specify the spatial location of the pulsing neuron N_7. The AER interface

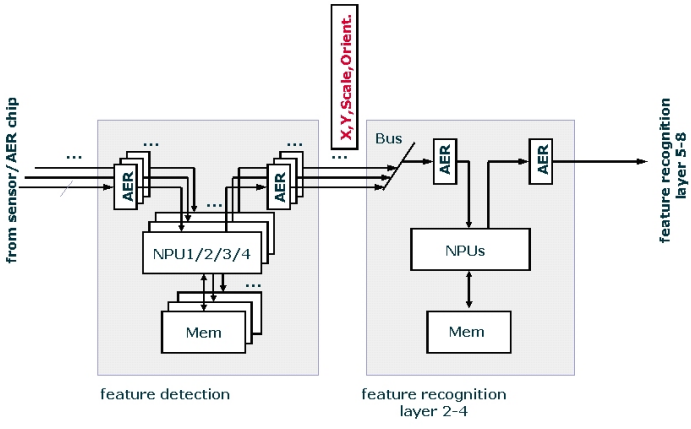


Fig. 17.15 Preliminary scheme of coupling a feature detection unit with a feature recognition unit. The layers $2-4$ are responsible for processing the operations O1 and O2.

contains FIFOs with sufficient memory capacity to supply the recognition circuit with pulses at a constant rate.

Each incoming pulse event is to be checked whether the sending neuron N_7 is connected to the zone or not. Note that a zone is specified by a set of N_7 neurons. Each of these neurons is defined by a unique 26-bit index. The index of an incoming pulse event is compared to the set of 26-bit words which define a zone. If any index of this set matches the incoming index, the pulsing N_7 neuron is considered to belong to the zone. In this case the operations O1 as well as O2 have to be executed.

At the start of the operations O1/O2, the membrane potential of N_a will be updated. If the neuron N_a stays quiescent (i.e. does not fire) and N_3 is not precharged at the same time, the adjacent neuron N_4 has to fire. This pulse is immediately forwarded to the synchronisation layer. Conversely, if either N_a fires or N_3 is precharged, N_4 has to stay quiescent. Furthermore, when N_a fires, the membranes of all N_3 neurons have to be precharged. The precharge operation represents a critical path since all 480 gradient detectors of a zone have to be updated. Similar to the detector architecture, the precharging states of the N_3 neurons will be stored in a dedicated memory which allows multiplexed processing. In the simplest case a single NPU is shared by all gradient detectors. The update of a particular gradient detector takes four clock cycles:

T1: generation of the target address for access to precharge state,
T2: readout of precharge state from memory,
T3: update of precharge state using an NPU,
T4: rewrite of precharge state into the memory.

We are interested in the required clock frequency of the NPUs. As in the case of the feature detector, the clock frequency depends on the incoming pulse event rate. As already outlined in previous sections, an event rate of 314.57 Mevents/s is to be expected on the input side. The estimated event rate is almost independent of the segmentation. If a given pulse arrives at a zone, the whole set of zone-specific indices have to be scanned until the next pulse event arrives at the input side. In addition, all 480 gradient detectors have to be updated within the same time frame. If we make the assumption that all operations involved are carried out sequentially, $480 + 4 \times 480 = 2400$ clock cycles have to be allowed for between consecutive pulse events. Theoretically, this can be done with a clock frequency of $755\,GHz$. Again, several subtasks have to be parallelized in order to sufficiently downscale the clock frequency as well as the power consumption.

Scanning the Index Set

Fortunately, it is possible to scan the index set in parallel. So-called CAM-arrays (CAM = *content addressable memory*) are well suited for a parallel comparison of a bit vector with a set of indices (Figure 17.16). The individual n-Bit indices will be stored in an array. In addition to the memory cells, for every index an n-bit comparator is integrated, which on the one hand is connected to the local memory cells and on the other hand is linked to the input vector. Hence, a parallel comparison of the input vector with the set of stored indices is achieved.

If a particular index matches the given input word, two output signals become active. These signals activate two priority encoders which are located at the boundary of the CAM array. First, a hit-signal is generated by the priority encoders. Then both priority encoders encode a data word which uniquely specifies the location of the match. If the indices are cleverly assigned to the cells, the location of the total match equals the index of the output neuron N_4 of layer 4 (Figure 17.14). At configuration time it is important

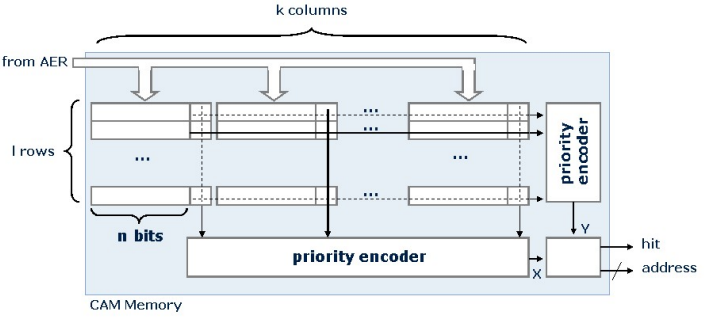


Fig. 17.16 CAM memory for the comparison of n-bit words with a given n-bit input vector. Each cell of the $k \times l$ array contains n memory cells as well as a n-bit comparator.

that all indices of a zone are distinct, so as to prevent multiple responses from different cells.

The architecture for a CAM cell is shown in Figure 17.17. It is possible both to store an index as well as make a comparison with an n-bit index. The proposed circuit requires less than four clock cycles for a full comparison. A memory cell of a CAM cell has a 1.5 times larger cell than a standard 6T SRAM cell, due to the additional circuit for comparison, and because of the additional wiring. If we assume an overall number of 480×26 bits per zone in order to store 480 indices per zone (n=26), the occupied CAM area equals an area which is required for 18720 SRAM cells. If the architecture is specified by the parameters k and l, the additional area requirements for the priority encoders as well as for the CAM cell logic (excluding the n-bit comparator) equals $11 \times (k+l) + 1.5 \times (k \times l)$ logic gates. In contrast to the priority encoder circuits used for the AER chip, the registers and the multiplexer are not needed here.

In order to get the most compact layout possible for the CAM array, the aspect ratio (ratio between height and width) should be between 2 and 0.5. This can be achieved by choosing $l = 96$ and $k = 5$, for instance. Based on this assumption, the number of logic gates for priority encoders, as well as for the X-Y logic, can be estimated at 1831 gates. Supplementary driver circuits are required to adjust the capacitive load (the so-called fan-in) for the input signals. Each bit of the input vector is connected to 5×96 inputs within the CAM array. Here, for each input bit an equivalent of 60 logic gates has to be provided. The overall number for the driver circuits can be estimated to be $60 \times 26 = 1560$ logic gates.

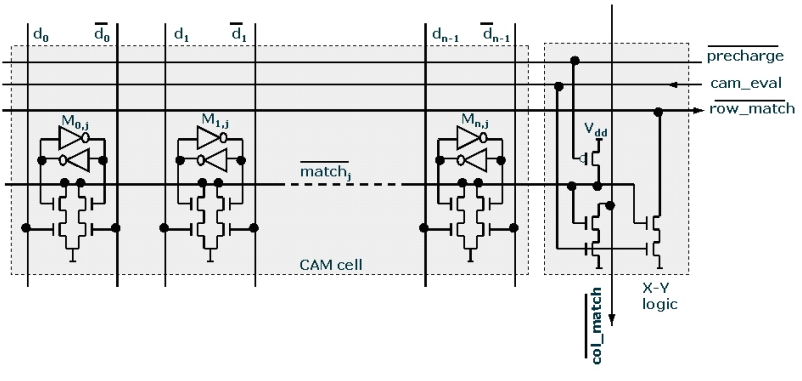


Fig. 17.17 Illustration of a CAM cell containing n memory cells and a n-bit comparator as well as some logic used for the generation of the hit-signal. The proposed circuit requires four clock cycles for a full comparison: (i) precharge of the signals $match_j$, col_match, row_match, (ii) apply input vector d_i, (iii) activation of the signal cam_eval, and (iv) evaluation of the signals col_match and row_match.

NPU Architecture

As already discussed, the update of the N_3 neurons located in layer 3 (Figure 17.14) is critical. It is possible to parallelize these operations. In the scheme proposed so far the memory interface is active in every second clock cycle. We add a second NPU to the architecture, which operates with a delay of one clock cycle in order to fully exploit the available memory bandwidth. Each NPU incorporated in the architecture processes a share of the overall set of N_3 neurons. Yet $4 \times 240 + 1 = 961$ cycles are necessary to process the complete set of N_3 neurons, which means that a clock frequency of at least $300\,GHz$ is required. This is far beyond a realistic implementation. Nevertheless, the number of clock cycles can be further reduced by continued parallelizing of operations. If we assume an architecture which in general allows us to process u N_3 neurons in parallel (i.e. $u/2$ N_3 neurons for an individual NPU, provided two NPUs operate in interleaved fashion), we get the numbers shown in Table 17.4. Even for the maximum degree of parallelism a feasible architecture is not obtained.

u	2	60	80	120	160	240	480
Number of cycles (N_3 precharge)	961	33	25	17	13	9	5
NPU gates [Gates] (1 zone)	346	1 854	2 374	3 414	4 454	6 534	12 774
NPU gates [MGates] (10.000 zones)	3.46	18.54	23.74	34.14	44.54	65.34	127.74

Table 17.4 Number of clock cycles and number of required logic gates for different degrees of parallelism

One way to reduce the clock frequency further is to reduce the rate of incoming pulse events. As in the case of the feature detector, the principle of segmentation can be applied. Subsequently we carry over the segmentation as well as the clock frequency from the feature detection layer to the recognition layer. Instead of forwarding all pulse events from the feature detection layer to a central bus (as shown in Figure 17.15), pulses of particular zones will be forwarded to individual zones (Figure 17.18).

Particular zones show an overlap of up to 9 segments. The overlap depends on the size of the segments, as well as on the location of the zone. For this reason pulse events from neighbouring segments are merged. In Figure 17.19 an architecture is shown which merges pulse events from four neighbouring segments. The MERGE unit contains four FIFOs. These FIFOs are read-out by the multiplexer in an alternating way.

With equal clock frequencies for the NPUs of the feature detector and the feature recogniser, separate circuits for clock synchronisation can be omit-

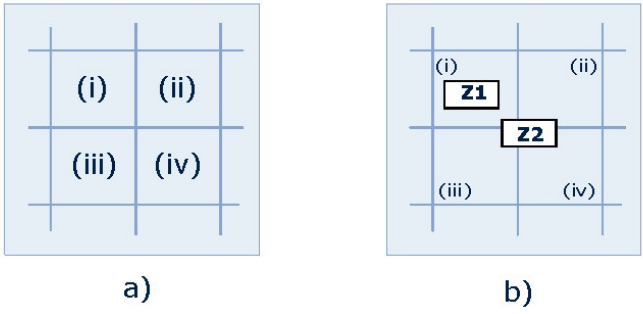


Fig. 17.18 a) Detailed view of a segmentated architecture consisting of segments (i), (ii), (iii) and (iv) b) two different zones (Z1, Z2) which cover the segments. Zone Z1 receives pulses from segment (i), zone Z2 receives pulses from segments (i),(ii),(iii) and (iv).

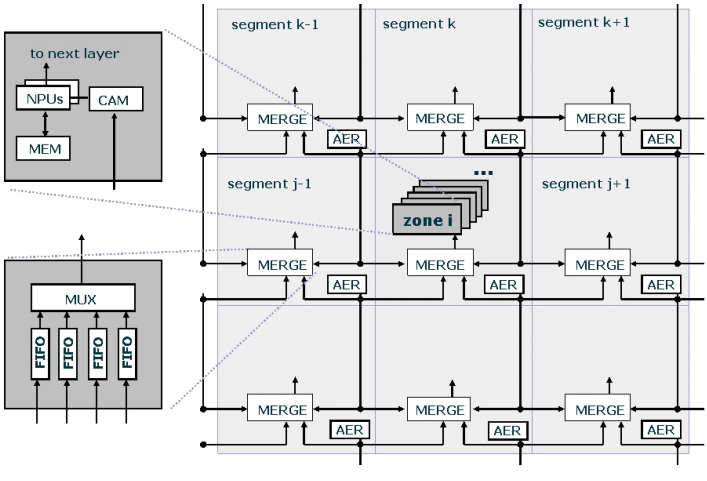


Fig. 17.19 Combination of pulse events which originate from four neighbouring segments. The MERGE unit receives pulse events on the input side and is responsible for the formation of an output stream of pulses which is forwarded to individual zones. The AER unit receives pulse events from the feature detection layer via the 3D interface.

ted. Then, for any given segmentation and for any given clock frequency, the number of available cycles n_T to fully process a zone are given in Table 17.5.

If we assume four clock cycles for the evaluation of the CAM array, we can determine the minimum degree of parallelism as well as the number of required logic gates. Table 17.6 summarises the system complexity. The results depend on the outcome presented in Table 17.4. An optimal area utilization

segments	1	64	256	1024	4096
Event rate per segment f_E [MEvents/s]	314.57	19.66	4.92	1.23	0.69
FIFO size per segment [kBytes]	3 328.00	208.00	52.00	13.00	7.31
FIFO area per segment $[mm^2]$	9.75	0.61	0.15	0.04	0.02
clock frequency detector f_D [MHz]	16 043.21	274.73	74.97	22.09	7.41
$n_T = f_D/f_E$	51	13	15	17	10

Table 17.5 Event rate at the output side of the MERGE unit as well as FIFO size and n_T as a function of the size of the segments.

Segments A-scale	1	64	256	1024	4096
clock frequency [MHz]	16 043.21	274.73	74.97	22.09	7.41
available cycles $n_T - 4$	47	9	11	13	6
u_{min}	<60	240	240	160	480
NPU equivalent gates per zone	<1 850	6 534	6 534	4 454	12 774
NPU area $[\mu m^2]$ per zone	<1 295.0	4 573.8	4 573.8	3 117.80	8 941.8
NPU area $[mm^2]$ 10 000 zones	<12.95	45.74	45.74	31.18	89.42
CAM + memory Area $[mm^2]$ 10 000 zones	92.40	92.40	92.40	92.40	92.40
FIFO area $[mm^2]$ scales A-E	48.75	52.41	51.95	51.95	116.89
3D-Via area $[mm^2]$	2.5025	2.5344	2.6194	2.9092	3.8640
Overall area $[mm^2]$	<156.60	193.08	192.71	178.44	302.57

Table 17.6 Effort of feature recogniser, layers 2-4.

is obtained by sub-dividing layer A into 1024 segments. In this case it is sufficient to realise the feature recogniser (i.e. layers 2-4 of the recogniser) by 3 chip layers. This architecture can deal with 10 000 different zones.

In Figure 17.20 the integration of 3 chip layers into the stack architecture and the required number of 3D vias are schematically illustrated. The wiring between layer 4 and layer 5 of the zone architecture is implemented as a hard-wired connection. Since each zone contains 20×24 neurons on the output side, $5 + 5$ bits are sufficient to specify the receiving neuron in layer 5.

Feature Synchroniser

The feature synchroniser includes 20×24 neurons on the input side (layer 5, Figure 7.8). If we take the architecture shown in Figure 7.8 as a basis, the synchroniser includes 480 neurons as well as 6168 dynamic synapses. Additionally, if we assume an architecture which is divided into 20 rows, we obtain 980 dendritic synapses as well as 20 dynamic synapses and 120 neurons for the detection of synchronisation. From practical results regarding the spot detector, it is possible to estimate the required area for the analogue circuit parts. During the changeover from 130 nm to 40 nm, the shrink factor is about 4

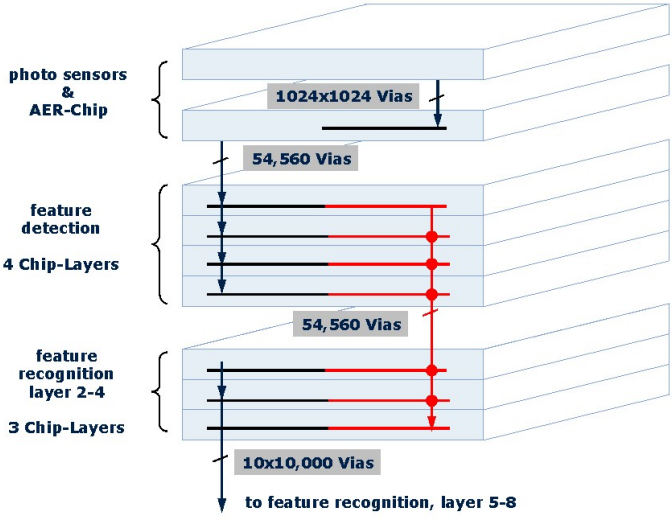


Fig. 17.20 Detailed view of the cascade

and differs significantly from the shrink factor obtained with digital circuits (approx. 10). If identical operating points for the analogue circuits (designed in subthreshold circuitry) are chosen, we have to take into account that signal currents increase by a factor of 2. This is caused by a slightly diminished threshold voltage for 40 nm (compared to 130 nm).

Based on these results, the capacities used for dynamic synapses as well as for the representation of membranes increase by about a factor of 1.26, which causes time constants to decrease by a factor of 0.63. Obviously the circuit design parameters for the analogue circuit parts have to be adapted if we aim to keep time constants as in 130 nm. Hence, only rough area estimations are possible for 40 nm. The largest area savings at the crossover from 130 nm to 40 nm are expected for dynamic synapses: for the synchronisation detector the dynamic parameter μ is enlarged by a factor of 16.67 (spot detector: $\mu = 0.3ms^{-2}$, synchronisation detector: $\mu = 5ms^{-2}$). In this case we shrink the capacity, which stores the current weight, as much as possible and simultaneously keep the current for weight adaptation as small as possible. Then, for the 40 nm re-design, significant area savings can be expected since the capacity for storing the weights occupies a considerable amount of area.

If we still assume that the current for weight adaptation is increased by a factor of 2, the area required for the capacity for the weight storage is given by $16.67/(2 \cdot 1.26) = 5.59$. Table 17.7 gives area estimations for particular components in 130 nm as well as 40 nm. Table 17.8 presents projections of the area requirements of a zone including 20×24 pixels.

	Neuron	Synapse $\mu = 0.3\,ms^{-2}$	Synapse $\mu = 5\,ms^{-2}$	AER circuit	pulse recovery
Fläche [μm^2] (130 nm)	423	212	106	240	200
Fläche [μm^2] (40 nm)	105.09	52.67	26.33	22.72	18.93

Table 17.7 Area requirements for subcircuits in 130 nm and 40 nm

Technology	no. neurons	no. synapses	AER-FIFO [kByte]	Area [mm^2] neurons	Area [mm^2] synapses	Area [mm^2] digital periphery	area [mm^2] overall
130 nm CMOS	600	7 168	0.59	0.25	0.76	0.096	1.13
40 nm CMOS	600	7 168	0.59	0.06	0.19	0.009	0.26

Table 17.8 Estimated area requirements of a zone consisting of 20×24 pixels

In this chapter, we assume an architecture capable of recognising 1 000 objects, i.e. 10 000 zones have to be implemented. Each particular zone has a dedicated AER interface (which also has to be implemented for each individual zone) and requires 10 3D vias for the connection of the feature recogniser and the synchroniser. The overall number of 3D vias is given by $10\,000 \times 10 = 100\,000$. This number is in the same order of magnitude as the number of 3D vias required for the connection of the feature cascade and the feature recogniser. If the overall chip area is limited to $8 \times 8\,mm^2$, no more than 200 zones can be integrated on one chip, as $53.2\,mm^2$ are required for the integration of logic (analogue/digital) already. The residual area is occupied by the landing pads of the 3D vias. Altogether, 50 chips are necessary for the integration of 10 000 feature synchronisers (Figure 17.21).

17.6 Summary

The vision cube for recognition of 1000 objects is composed of the following chips:

- 1 imager with 1000x1000 pixels and 1 million vias,
- 1 chip for resolution pyramid and pulse encoding with 54560 vias,
- 4 chips for feature detection and programmable orientation, each with 54560 Vias,
- 3 chips for definition of zones, each with 100.000 vias,
- 50 synchroniser chips for feature recognition.

Except for the imager and the synchroniser chips, all chips are designed using digital circuitry.

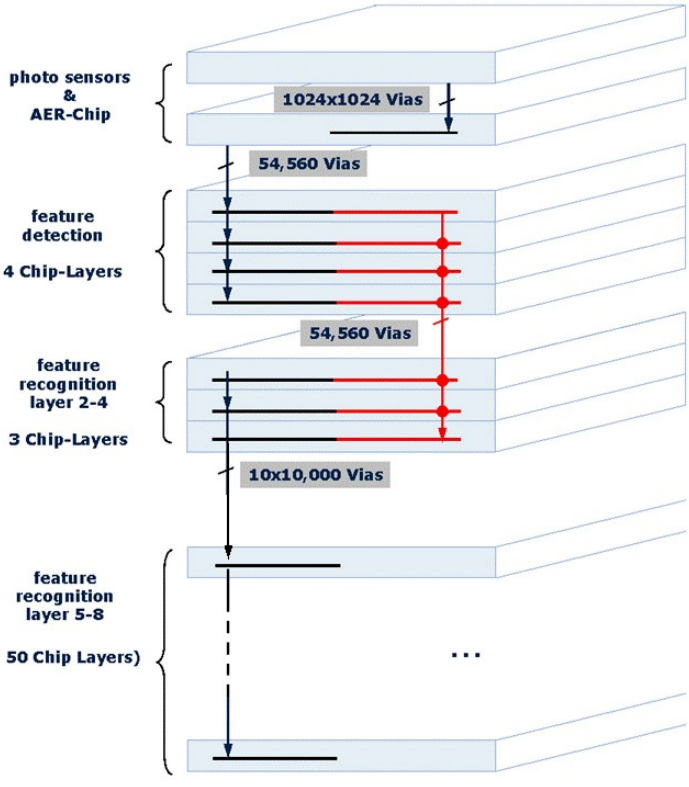


Fig. 17.21 Vision cube capable of recognising 1000 objects

A characteristic feature of the vision cube is the high degree of parallelism:

- more than 1 million pixel cells including analogue-pulse converters, as well as rate dividers,
- about 175.000 NPUs for the implementation of the feature cascade,
- 10.000 synchronisation detectors for the feature recogniser
- and finally about 6.95 million vias.

A total of 59 chips is assembled in the chip stack. The physical height is given by $\approx 0.9\,mm$ and the largest area of $\approx 1\,cm^2$ is occupied by the imager chip. The other chips in the stack occupy an area of $\approx 65\,mm^2$. The power consumption has been estimated to be less than $10\,W$. In total, the vision cube contains $\approx 90\,MB$ memory and 240 million gates.

An alternative implementation would be given by a network of standard general-purpose CPUs. If we aim for a real time feature cascade without the feature recognisers, ie. the first 6 chip layers of the stack, more than 4 computers (2 GHz class) have to be provided and cross-linked. As the robust recognition performance provided by the feature recognisers is hard to esti-

mate in terms of algorithms, let alone MIPS and memory, we leave it to the reader to answer how many computers would really be needed to replace 50 analogue chips.

It appears that 3D technology is indispensable for the construction of artificial brains. From cost estimations it has been concluded that in the start-up period 3D technology ready for mass production is about 30% more costly compared to state-of-the-art packaging technologies. On the other hand, 3D technology is accompanied by a greatly improved reliability of the system (containing 60 chips). This complexity can be compared to today's car electronics which is composed of up to 1 hundred microcontrollers.

18

End

If we estimate 10^{12} neurons and 10^{15} synapses for the human brain, then about 0.14 percent of all neurons are necessary for the cascade of feature detectors/recognisers for 1000 objects, as described here. For the recognition of one million objects, we would have to reckon with $31 \cdot 10^9$ neurons, (3.1 percent) and $135 \cdot 10^9$ synapses (0.013 percent). After all, if we imagine the one million objects related to each other by means of a gigantic correlation matrix, then another 10^{12} synapses would be necessary. This would, however, only represent one thousandth of all synapses. Hence, our vision architecture marks nothing more than a first step to understanding the brain.

On the other hand, to create a neural architecture for feature detection and robust head recognition as well as the subsequent reconstruction of the retina image on the sole basis of pulse neurons and dynamical synapses, we believe, is not a small accomplishment. Taking into consideration the multitude of effects developed by the simple neuron and synapse model in the course of building basic information processing structures, it appears that the construction kit is sufficiently filled to start the design of a comprehensive object world.

There is, therefore, a lot to research and this could be grounds for rejoicing. On the one hand there are many applications for intelligent machines, which release people from strenuous or dangerous tasks. On the other hand, we are at the beginning of a road which could lead to artificial brains being more efficient than the human brain. Along this path robots could be produced to replace soldiers and as a result, when the technology has become cheaper, individually tailor-made service robots could appear on the market for the man in the street – the efficiency of killing on a large scale and social isolation could rise to levels never previously imagined.

Nature has endowed us with curiosity so that research on artificial brains cannot be halted. However, since Hiroshima we have lost our innocent curiosity and inquisitive play. In the end we are always responsible – but do we know what we are doing?

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 325–326, 2010.
© Springer-Verlag Berlin Heidelberg 2010

Perhaps the research on the brain assumes more mature human beings than we are at present: those professionals who are quite content with dealing with or explaining highly specialised connections. Instead, we would have to restrain the professional and develop within ourselves a person who is responsible to society – this, at any rate, is possible. A more balanced approach would certainly see our research arrive at a different destination.

19

Appendix

19.1 Simulator for Chapters 3, 4

Jens Harnisch

The analytical model describing the macrodynamic behavior of the pulse coupled neurons was implemented in a C/C++ simulator.

The major focus for the implementation was an easy-to-understand and maintain source code. There are certainly several options to optimize run time and memory consumption. However, as the theory to describe the behavior of the pulse-coupled neurons was developed over a longer time span, the ease of maintenance was the primary criterion for the implementation. The reader of this book is welcome to reuse parts of the source code or the complete package in his own simulations.

For many simulations in this book, fully connected neurons were used, without back coupling to the neuron itself. In this case the amount of required memory to store the weights between the neurons expands quadratically with the number of neurons. When calculating the analytical model of the neuron behavior, there is an additional problem: For n neurons there exist 2^n possible firing patterns, which means the required memory grows exponentially with the number of neurons. Assuming the likelihood of a firing pattern is stored in a float value of 4 Byte, we would need, for instance, 4 GByte to store the firing patterns of a net with 30 neurons. Certainly one might use hard disk space to store the firing patterns, or apply other tricks. However, this usually leads to increased calculation time, and only moderately increases the number of neurons.

Handling of Numerical Problems

When calculating the theoretical model, several numerical problems need to be considered, in particular when calculating numbers with widely different

U. Ramacher and C. v.d. Malsburg: On the Construction of Artificial Brains, pp. 327–346, 2010.
© Springer-Verlag Berlin Heidelberg 2010

exponents. It is strongly recommended to check whether the results of the calculations are numerically valid, as in the following example.

```
#include <float.h>

if (!(_finite(result) && (_fpclass(result)!=_FPCLASS_QNAN)
{
    cout << "this is not a valid result" << endl;
    ...
    // call error handler of the simulation
}
```

The function `_finite()` is returning `TRUE`, in case the argument x is not infinite, hence $-INF < x < +INF$ is valid.

`_finite()` returns `FALSE`, in case the argument is infinite or a `NaN` value (Not a number). Possible results are:

```
_FPCLASS_SNAN   NaN
_FPCLASS_QNAN   Quiet NaN
_FPCLASS_NINF   Negative infinity (-INF)
_FPCLASS_NN     Negative normalized non-zero
_FPCLASS_ND     Negative denormalized
_FPCLASS_NZ     Negative zero (-0)
_FPCLASS_PZ     Positive 0 (+0)
_FPCLASS_PD     Positive denormalized
_FPCLASS_PN     Positive normalized non-zero
_FPCLASS_PINF   Positive infinity (+INF)
```

For the simulations it was found sufficient to check the floating point class for `_FPCLASS_QNAN`. The functions `_finite()` and `_fpclass()` are provided by Microsoft Visual C++.

Details of Simulation

The simulation was implemented with the help of an event driven simulator, essentially working in two phases:

- In the first phase, the point in time of the next event is calculated for all neurons, that is, the point at which the neuron starts or stops to fire, based on the current status of firing and non-firing neurons.
- In the second phase, from the set of possible events calculated in the first phase, the closest in time is picked. The other events will be ignored. The new status of all neurons is calculated, still based on the old constellation of firing and non-firing neurons. Only after this calculation is completed will the neurons change their status, if applicable. The calculation starts again with the first phase.

More advanced simulation techniques can be found in the literature. There are several options to save some calculations, e.g. by storing event lists, or by running the simulation in parallel on several computers. On the other hand, those advanced techniques lead to more demanding test efforts and do not necessarily improve the maintainability of the source code. For the simulations in this book it turned out to be better to rely on a simple simulation approach. Furthermore this simple approach made it easier to compare the simulation results with those from other simulators. The way the whole program works, including the calculation of the analytical model, is depicted in the following Figure 19.1.

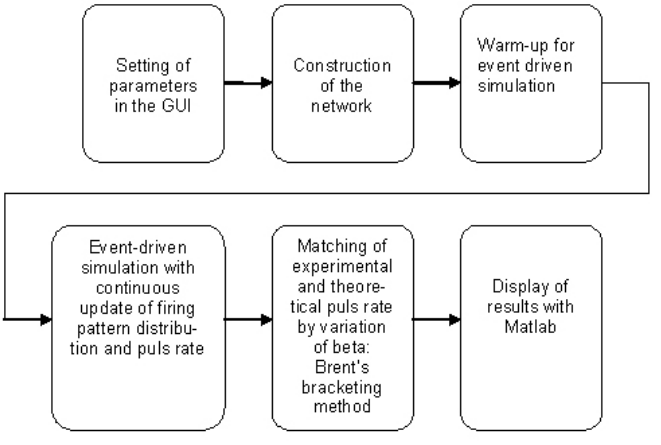


Fig. 19.1 Program flow for the simulation and calculation of the analytical model describing the neural net behavior

Integration with MATLAB

The easy integration of the program with MATLAB turned out to be particularly helpful. Different types of diagrams could be easily generated.

For using MATLAB a dedicated header file from MATLAB needs to be included (engine.h), we need to link with libraries from MATLAB (libmx.lib, libmat.lib, libeng.lib) and to generate a valid pointer onto a MATLAB engine. When using the MATLAB engine, it is only necessary to copy the values from the simulation or analytical calculations into a shared memory space, and via strings commands are sent to MATLAB, for displaying the diagrams.

A simple source code example is given below, displaying the array arg1 with count values in a diagram. On the x-axis the index of the measurement

values is shown. The y-axis shows the values of the array arg1. A slightly modified version of the function was used for instance to display the likelihoods of the firing patterns in MATLAB.

```
void matlab(double* arg1, unsigned long count)
{
    Engine *ep=NULL;
    mxArray *mx_index = NULL;
    mxArray *mx_vals  = NULL;
    double* index_ptr = NULL;
    double* val_ptr   = NULL;
    unsigned long cnt;

    // start Matlab
    if (!(ep = engOpen("matlab")))
    {
        fatal("can not start MATLAB engine");
    }

    index_ptr= new double[count];
    if (!index_ptr)
    {
        fatal("memory allocation failure");
    }

  // we define the x values, and simply assume an
  // index starting with 0
    for (cnt=0; cnt < count; cnt++)
    {
        index_ptr[cnt]= (double) cnt;
    }

    // we copy the values, this may not be necessary
    val_ptr= new double[count];
    if (!val_ptr)
    {
        fatal("memory allocation failure");
    }
    for (cnt=0; cnt < count; cnt++)
    {
        val_ptr[cnt]= arg1[cnt];
    }

    // we define variable in the Matlab environment
    mx_index = mxCreateDoubleMatrix(1, count, mxREAL);
```

```
    mxSetName(mx_index, "mx_index");
    mx_vals = mxCreateDoubleMatrix(1, count, mxREAL);
    mxSetName(mx_vals, "mx_vals");
    engEvalString(ep, "figure();");

    // now we actually copy the values into Matlab
    memcpy((void *)mxGetPr(mx_index), (void *)index_ptr,
         sizeof(double) * count);
    memcpy((void *)mxGetPr(mx_vals), (void *)val_ptr,
         sizeof(double) * count);

    // place the variable array into the MATLAB workspace
    engPutArray(ep, mx_index);
    engPutArray(ep, mx_vals);
    engEvalString(ep, "figure(1);");

    // plot the result
    engEvalString(ep, "h=semilogy(mx_index,mx_vals);");
    engEvalString(ep, "set(h,{'LineStyle'},{'-'};)");

    // annotations for the diagram
    engEvalString(ep, "title('probabilities for
                              firing patterns');");
    engEvalString(ep, "xlabel('probability');");
    engEvalString(ep, "ylabel('index of firing pattern');");

    // use engOutputBuffer to capture MATLAB output
    engOutputBuffer(ep, buffer, sizeof(buffer));

    mxDestroyArray(mx_index);
    mxDestroyArray(mx_vals);

    // close the Matlab
    engEvalString(ep, "close;");
}
```

Working with the Simulator

The simulator is provided with a very simple graphical user interface for MS Windows (cf. Figure 19.2). The user can set the net parameters (e.g. number of neurons), the simulation parameters (e.g. number of simulated events) and the parameters for the calculation of the analytical model (e.g. range which is to be searched for a matching beta).

The parameters are explained in detail below:

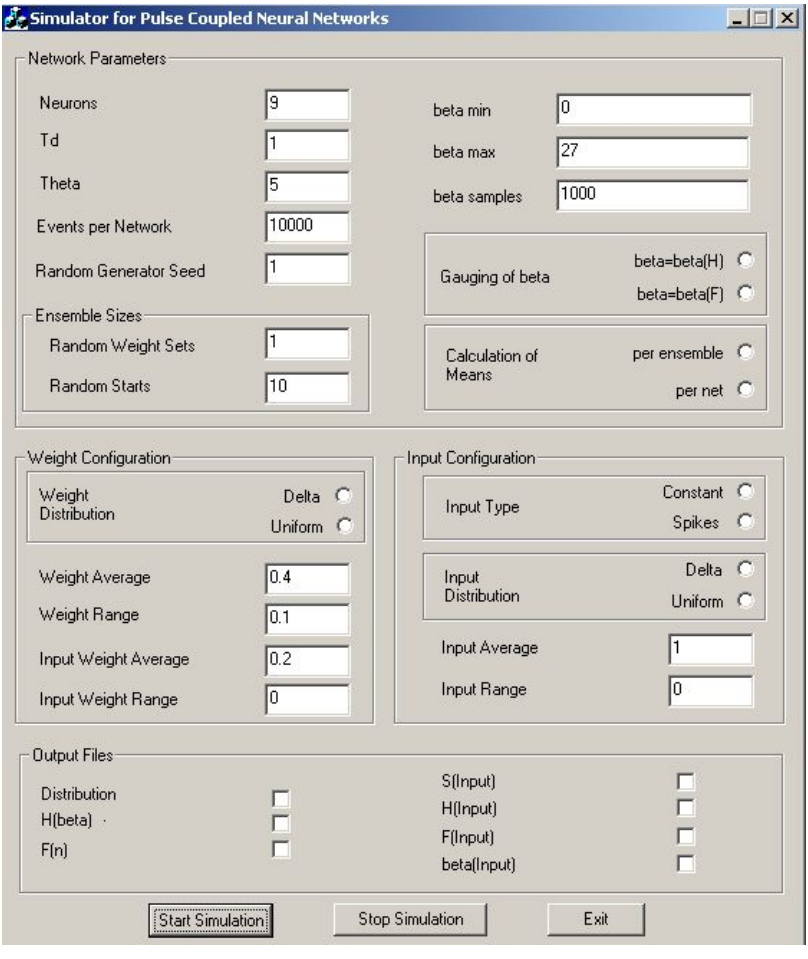


Fig. 19.2 Graphical interface of the simulator

Network Parameters

- *Neurons*: Number of neurons in the fully connected net
- *Td*: Duration of a pulse produced by a neuron
- *Theta*: Neuron threshold
- *beta min und beta max*: Search interval for the calculation of beta
- *beta samples*: This parameter does not influence the analytical model. However, some experiments demonstrate various functions dependent on beta. For those diagrams, the number of beta samples shown on the x-axis, needs to be given.
- $beta = beta(H)$: Calculation of beta based upon the measured average value H of the signal energy h

- *beta = beta(F)*: Calculation of beta based upon the average puls rate F of a net
- *Events per Net*: Total number of events per net in a configuration
- *Random Generator Seed*: The weights and accumulators can be initialized with random numbers. The sequence of those random numbers depends on the random generator seed. With the same seed the simulation results are absolutely reproducable. With a different seed on the other hand, more or less different simulation results can be induced. Because the net sizes which can be simulated are rather small, and hence particular weights can significantly influence the simulation results of single nets, we recommend experimenting with different seeds. Calculations for ensembles are not affected by this parameter.
- *Ensemble Size – Random Weight Sets*: The corresponding experiment will be run automatically with different weight initializations, if a number larger than 1 is given.
- *Ensemble Size – Random Starts*: The corresponding experiment will be run automatically with different initializations for the membrane potentials and completed firings.
- *Calculations of Means*: „per net“ = Entropy, average energy and beta are calculated per net, and then average values are calculated across the ensemble. „per ensemble“ = Entropy, average energy and beta are calculated across the ensemble.

Input Configuration

- *Input Average*: Average value of the external stimulus
- *Input Range*: Range of the external stimuls. For example, if the stimulus average is set to 0.5, and the stimulus range to 0.5, then the external stimuli are uniformly distributed in the range [0,1].
- *Input Type*: „constant“ = Input constant in time. If „spikes“ is selected, then pulses are fed into the net, with the time average of the pulses corresponding to the number chosen for *InputAverage*.
- *Input distribution*: „delta“ = Input known. „uniform“ = Average and range known for uniform distribution of input pulses.

Configuration of Weights

- *Weight Average*: Average of all weights
- *Weight Range*: Range of the neuron weights. For example, if the average weight is set to 0.5, and the weight range to 0.2, then the weights are uniformly distributed in [0.3, 0.7].
- *Input Weight Average*: Average value of the weights to the external stimuli
- *Input Weight Range*: Range of the weights to the external stimuli. For example, if the average input weight is set to 0.3 and the input weight range to 0.2, then the weights are uniformly distributed in [0.1, 0.5].

- *Weight Distribution*: „delta“ = Weights known. „uniform“ = Average and range for uniform weight distribution is known.

Output Files

- *distribution* = Histogram and distribution of the firing patterns are shown; additionally for entropy, average energy and the pulse rate.
- $H(beta)$ = The average measured value of H and the theoretical average value of H(beta) are shown as a functions of beta. This serves for checking the matching procedure.
- $F(n)$ = Pulse rate for each neuron is shown.
- $S(inp)$ = The net entropy is calculated from the experiment and the theory, and is shown as a function of the inputs.
- $H(inp)$ = The average energy of the net is calculated from the experiment and the theory and is shown as a function of the inputs.
- $F(inp)$ = The average puls rate of the net is calculated from the experiment and the theory and and shown as a function of the inputs.
- $beta(inp)$ = beta is shown as a function of the inputs.

Control commands:

- *Start simulation*: This command starts the simulation, and following the simulation, automatically evaluates the analytical model. The results are shown in MATLAB. Depending on the parameters chosen, the calculation can take from a few seconds up to several days and longer. At the beginning of the simulation, the simulator generates a short beep; a long beep is generated at the end.
- *Stop simulation*: This command stops the current simulation. The parameters chosen are maintained. Now we can modify some parameters, e.g. shorten the simulation time.
- *Exit*: This command ends the program. MATLAB diagrams which are already present are maintained. Values, which were used in MATLAB, are maintained as well and can be used to generate new diagrams.

The source code and the program are available upon request.

19.2 Axon Model by Hodgkin and Huxley

Jörg Schreiter

In 1952, Hodgkin and Huxley published a model that describes how action potentials develop and travel in the extremely long axons of squid. The model is a set of state equations for a patch of axon membrane. A clear presentation of the model can be found in [65]. Important parts of it are depicted in the following.

A part of the model can be represented by the electrical circuit of Figure 19.3. The patch of membrane considered, has the area A and forms a capacitor with capacitance

$$C = 1\frac{\mu\mathrm{F}}{\mathrm{cm}^2}A\ . \tag{19.1}$$

Across that capacitor the membrane voltage V can be measured, which is one state variable of the model. The Hodgkin-Huxley model contains voltage controlled ion channels, carrying $Na^+$and K^+ions. These ion channels on the membrane patch under consideration are modelled as controlled resistors with conductances G_{Na} and G_{K}, respectively. All other ion channels of the membrane patch are combined and modeled as a resistor with fixed conductance G_{L}. The ion concentrations in the axon's cytoplasm and in the extracellular medium stay almost unchanged, when ionic currents flow through the ion channels. Therefore, the reversal voltages V_{Na}, V_{K} and V_{L} can be assumed to be constant.

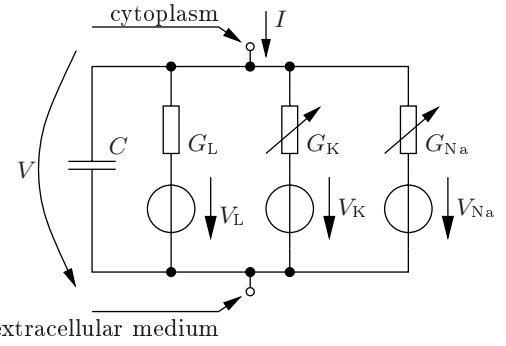


Fig. 19.3 Electric part of the Hodgkin-Huxley model.

The conductances G_{Na} and G_{K} reach their respective maximum values $\widehat{G}_{\mathrm{Na}}$ and $\widehat{G}_{\mathrm{K}}$ respectively when all corresponding ion channels are open. However, in most cases some ion channels are blocked.

Each ion channel is considered to be built up from four units, which are either in a conducting or a non-conducting state, and which switch individually between the two. Na^+channels are thought to be composed of three identical units m and one different unit h, while K^+channels are thought to be built of four identical units n. An ion channel only conducts, when all its units are conducting. The relative number of conducting units is expressed in terms of state variables m, n, and h, which are dimensionless numbers.

The peak conductances $\widehat{G}$ as well as the Nernst-voltages V_{Na} and V_{K} of Na^+- und K^+-ions, respectively, can be found in [65]. Additionally, there is $G_{\mathrm{L}} = 0,36\,\mathrm{mS/cm}^2$. The complete Hodgekin-Huxley-Model can be expressed as an equation system:

$$
\begin{aligned}
C\dot{V} &= I + \widehat{G}_{\mathrm{Na}} m^3 h (V - V_{\mathrm{Na}}) + \widehat{G}_{\mathrm{K}} n^4 (V - V_{\mathrm{K}}) + G_{\mathrm{L}} (V - V_{\mathrm{L}}) \ , \\
\tau_0 \, \dot{m} &= \alpha_m(V)(1-m) - \beta_m(V) m \ , \\
\tau_0 \, \dot{n} &= \alpha_n(V)(1-n) - \beta_n(V) n \ , \\
\tau_0 \, \dot{h} &= \alpha_h(V)(1-h) - \beta_h(V) h \ .
\end{aligned}
\tag{19.2}
$$

Therein, time is scaled by a factor $\tau_0 = 1\,\mathrm{ms}$. The functions α_m, α_n, α_h, β_m, β_n and β_h represent the voltage dependent rates of state changes of the channels' units. Hodgkin and Huxley determined these functions empirically and fitted them to measurements as follows:

$$
\alpha_m(V) = \frac{-4 - V/10\,\mathrm{mV}}{\mathrm{e}^{-4 - V/10\,\mathrm{mV}} - 1} \qquad \beta_m(V) = 4\mathrm{e}^{-3{,}6 - V/18\,\mathrm{mV}} \ , \tag{19.3}
$$

$$
\alpha_n(V) = \frac{-0{,}55 - V/100\,\mathrm{mV}}{\mathrm{e}^{-5{,}5 - V/10\,\mathrm{mV}} - 1} \qquad \beta_n(V) = 0{,}125\mathrm{e}^{-0{,}8 - V/80\,\mathrm{mV}} \ , \tag{19.4}
$$

$$
\alpha_h(V) = 0{,}07\mathrm{e}^{-3{,}3 - V/20\,\mathrm{mV}} \qquad \beta_h(V) = \frac{1}{\mathrm{e}^{-3{,}5 - V/10\,\mathrm{mV}} + 1} \ . \tag{19.5}
$$

The Hodgkin Huxley Model can explain the time course of action potentials arising at the soma close to the axon. If an electric current pulse is applied to the electric part of the model, e.g. a square pulse $I(t) = I_0(H(t) - H(t - t_0))$ of magnitude I_0 and duration t_0, with H being the Heaviside function (11.4), then the membrane voltage V may show a short spike with a peak value of up to about $+35\,\mathrm{mV}$.

As in the biological ante type, the system shows threshold behaviour. When pulses of fixed duration t_0 are applied, action potentials will occur only if the magnitude I_0 is above a threshold I_{th}. Figure 19.4 displays the time of the membrane voltage for some values of I_0. Around the threshold, there is a continuous transition from a small stimulus to the full action potential.

A particular value is considered to be the threshold value I_{th}, if $I_0 = I_{\mathrm{th}}$, the peak of the spike appears at the last time point (cf. Figure 19.4, curve b). By doing so, the set of curves is split into two distinct groups. This represents a decision about whether a action potential will occur or not. The decision never takes longer than the spike delay exactly at the threshold.

Another detail of the curves supports the choice of considering that value as the threshold: After the peak, the membrane voltage has a minimum that is below the resting potential V_R. For all $I_0 > I_{\mathrm{th}}$ the value at the minimum is close to that which occurs in the case of a full action potential.

Close to the threshold, the sensitivity[1] is very high. In Figure 19.4, the curves displayed with dash-dot lines correspond to a small range of pulse magnitudes: $-7{,}1 \cdot 10^{-13} I_{\mathrm{th}} \leq I_0 - I_{\mathrm{th}} \leq 7{,}1 \cdot 10^{-13} I_{\mathrm{th}}$.

[1] In Figure 19.4, the spike's peak is at $\widehat{t}, \widehat{V}$. The sensitivity $S_{I_0} := \partial \widehat{V} / \partial I_0$ of the peak height w.r.t. the pulse magnitude can be used to express how much the curve changes in response to a change of the input.

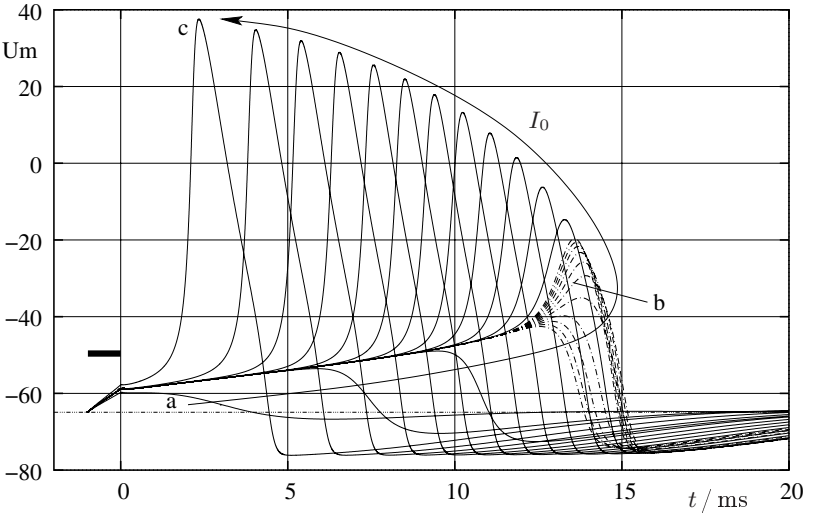


Fig. 19.4 Threshold behaviour of the Hodgkin Huxley Model when stimulated by an electric current square pulse of $t_0 = 1\,\mathrm{ms}$ duration and variable magnitude I_0. The pulse ends at $t = 0$ (thick solid line). At small magnitudes (a), the membrane voltage after a slight stimulus, quickly returns to the resting potential (dash-3-dots line). If the magnitude is above a threshold $I_0 = I_{\mathrm{th}}$ (b), then the membrane voltage shows a prominent peak. When the pulse magnitude is raised further, the peak of the spike quickly approaches the value of the full action potential (c). As indicated by a parameter arrow, the position of this peak changes, as I_0 changes. Curves are plotted according to the following set of values:
solid line: $\frac{I_0 - I_{\mathrm{th}}}{I_{\mathrm{th}}} \in \{-1,4 \cdot 10^{-1};\ -1,4 \cdot 10^{-5};\ -1,4 \cdot 10^{-9}\}$;
dash-dot line: $\frac{I_0 - I_{\mathrm{th}}}{I_{\mathrm{th}}} \in \{-7,1 \cdot 10^{-13};\ -5,6 \cdot 10^{-13};\ -4,2 \cdot 10^{-13};\ -2,8 \cdot 10^{-13};$
$-1,4 \cdot 10^{-13};\ 0;\ 1,4 \cdot 10^{-13};\ 2,8 \cdot 10^{-13};\ 4,2 \cdot 10^{-13};\ 5,6 \cdot 10^{-13};\ 7,1 \cdot 10^{-13}\}$;
solid line: $\frac{I_0 - I_{\mathrm{th}}}{I_{\mathrm{th}}} \in \{2,1 \cdot 10^{-12};\ 1,6 \cdot 10^{-11},\ 1,6 \cdot 10^{-10};\ 1,6 \cdot 10^{-9};\ 1,6 \cdot 10^{-8};$
$1,6 \cdot 10^{-7};\ 1,6 \cdot 10^{-6};\ 1,6 \cdot 10^{-5},\ 1,6 \cdot 10^{-4};\ 1,6 \cdot 10^{-3};\ 1,6 \cdot 10^{-2};\ 1,6 \cdot 10^{-1}\}$.

A threshold for the creation of an action potential can also be expressed in terms of the membrane voltage. If the membrane voltage V, in response to an input current, is above V_{th}, an action potential will occur a few milliseconds later. The voltage threshold value of the Hodgkin-Huxley Model is about $V_{\mathrm{th}} \approx -57\,\mathrm{mV}$. Actual creation of an action potential is dependent on the further time evolution of the membrane voltage, after it exceeds the threshold, in a complicated way.

19.3 Basic Transistor Circuits

Jörg Schreiter

Schalter und Ladungsinjektion

MOS transistors can be used as electrically controlled switches. The switch is considered to be turned on, if the gate voltage is above (NMOS) and below (PMOS), respectively, the drain and the source voltages by more than the transistor's threshold voltage:

$$\begin{aligned} V_G &> \max(V_D, V_S) + V_t \quad \text{(NMOS)} \ , \\ V_G &< \min(V_D, V_S) + V_t \quad \text{(PMOS)} \ . \end{aligned} \tag{19.6}$$

The channel is built up, when the transistor switch is turned on. The resultant charge in the channel flows from the drain and source terminals into the channel. When the transistor is turned off again, the channel charge is divided between the nodes connected to the drain and source terminals. Both effects are commonly designated as charge injections.

In many applications, charge injected at turn-on is equalized while an initial voltage difference between the drain and source terminals of the transistor switch decays. At turn off, the injected charge causes a voltage deviation at either drain or source, which is considered a voltage error.

The distribution of the channel charge very much depends on the switching speed. In the extreme case of a negligibly short turn-off, the channel charge is evenly distributed between the nodes connected to the drain and source terminals, and the voltages at the terminals change independently from each other.

At a very slow, quasi-static turn-off, the channel is still conducting while it is de-constructing. This allows continuous equalisation of the drain and source voltages. Thus, the channel charge is distributed, such that both terminals have almost equal voltages.

Below the surface charge in the channel, there is a depletion region in the bulk of the transistor [115, S. 142 ff]. The corresponding charge does not take part in the charge injection. Instead, the majority of the depletion charge flows through the bulk terminal.

Figure 19.5 shows charges and capacitances connected to the charge injection with the help of a cross-section of a modern integrated MOS transistor.

FigureMOS-switch shows a cross-section of an MOS transistor and the most important parasitic capacitors. The active area of the transistor, i.e. drain, source, and channel, is isolated from other transistors by the shallow trench of isolation oxide, that extends a few hundred nanometers below the silicon surface. On the left side, the creation and extinction of the channel charge

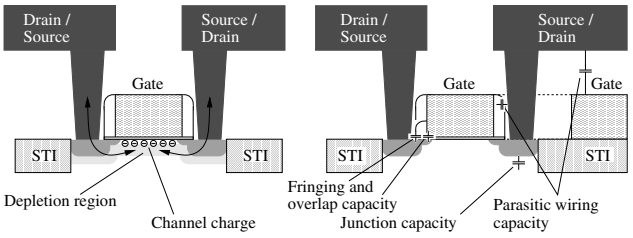


Fig. 19.5 Cross-section of an MOS transistor. See the text for an explanation of a charge injection. (STI: *shallow trench isolation*)

is shown. On the right side, capacitors symbolically indicate parasitic capacitances of the transistor, which are charged or discharged at the switching event and thus contribute to the charge injection. The junction capacitances of the drain and source diffusion areas do not directly contribute to the charge injection. However, they store a part of the injected charge and contribute to the final voltage change caused by the charge injection.

The switching transistor is now being used to control the charging of a capacitor C_2 according to Figure 19.6.

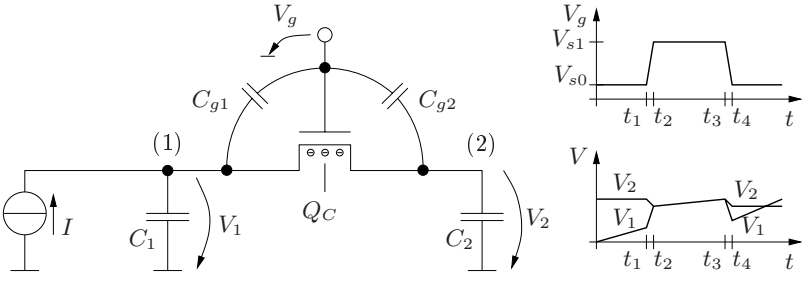


Fig. 19.6 Application of an switching transistor

It may be assumed that the remaining parts of the complete circuit, like other switches and a physical implementation of the voltage limited current source, may be neglected or have no influence at all. The gate terminal is connected to the switching voltage $V_g = V_s$, which takes the values V_{s1} for *on* and V_{s0} for *off*.

In the on-state, the transistor operates in the Ohmic region, and its drain-to-source voltage is negligibly small: $V_1 \approx V_2$. V_1 resp. V_2 is the source-to-bulk voltage. Because of the body effect, the transistor has an increased threshold voltage [115, S. 157 ff]

$$V_t = V_{t0} + \gamma \left(\sqrt{2\Phi_F + V_{SB}} - \sqrt{2\Phi_F} \right) \ , \tag{19.7}$$

where $V_{SB} = V_1$, V_{t0} is the threshold voltage at $V_{SB} = 0$, γ is the body effect constant, and $2\Phi_F$ is the surface voltage in strong version[2].

A minimum switching voltage $V_{s1} > V_1 + V_t$ is required, to put the transistor in strong inversion. Any upper limit of the switching voltage V_{s1} translates into an upper bound for the source and drain terminal voltages when the switch is conducting. E.g. V_{s1} is limited by the supply voltage $V_{s1} = V_{DD}$. Then V_1 and V_2 cannot exceed $V_{1,2} < V_{DD} - V_t = V_{DD} - V_{t0} - \gamma\big(\sqrt{2\Phi_F} + \gamma/2 - \sqrt{V_{DD} - V_{t0} - \gamma\sqrt{2\Phi_F} + \gamma^2/4 - 2\Phi_F}\big)$.

When the transistor is turned off, it is almost blocked. It still conducts an unwanted current, given by the weak inversion operation [92, section 3.5]:

$$I_D = \frac{W}{L}\beta U_T^2 \mathrm{e}^{\frac{V_G - V_{t0}}{nU_T}} \left(e^{-\frac{V_S}{V_T}} - e^{-\frac{V_D}{U_T}} \right) , \tag{19.8}$$

$$|I_D| \le \frac{W}{L}\beta U_T^2 \mathrm{e}^{\frac{V_{s0} - V_{t0}}{nU_T}} e^{-\frac{\min(V_1, V_2)}{U_T}} . \tag{19.9}$$

U_{s0} has to be chosen according to the maximum tolerable current $|I_{D,\max}|$ in the turned-off switch: $V_{s0} < n\big(U_T \ln(|I_{D,\max}|/I_{D0}) + \min(V_1, V_2)\big)$, where $I_{D0} = (W/L)\beta U_T^2 \exp(-V_{t0}/nU_T)$ is the drain current at $V_G = V_S = V_B$ and $V_D - V_B \gg U_T$, while $U_T = kT/q$ denotes the temperature voltage.

Just before the transistor is turned on at time t_1, the voltages V_1 and V_2 take the values $V_1(t_1)$ and $V_2(t_1)$. Later at time $t_{2'}$ ($t_2 < t_{2'} < t_3$), when the transistor has been turned on, the terminal voltages have been equalized. Now the common terminal voltage can be obtained assuming charge conservation:

$$\begin{aligned} & V_1(t_1)\big(C_1 + C_{g1}\big) + V_2(t_1)\big(C_2 + C_{g2}\big) - V_{s0}\big(C_{g1} + C_{g2}\big) \\ = {} & V_1(t_{2'})\big(C_1 + C_{g1}\big) + V_2(t_{2'})\big(C_2 + C_{g2}\big) - V_{s1}\big(C_{g1} + C_{g2}\big) \\ & + Q_C - \int_{t_1}^{t_{2'}} I(t)\, dt , \end{aligned} \tag{19.10}$$

where Q_C is the charge of the inversion layer forming the channel [115, S. 159]. It is given by

[2] The body effect is a change of the transistor's threshold voltage V_t, with reference to the gate-to-source voltage, when the source-to-bulk voltage V_{SB} changes. The constant $\gamma = \sqrt{qN_A 2\epsilon_0\epsilon_{\mathrm{Si}}}/C'_{\mathrm{ox}}$ is obtained from the substrate doping density N_A, the dielectric constant $\epsilon_0\epsilon_{\mathrm{Si}}$ of silicon, and the gate capacitance per square unit $'_{\mathrm{ox}} = \epsilon_0\epsilon_{\mathrm{SiO_2}}/d_{\mathrm{ox}}$. The surface voltage occurs between the silicon surface and the silicon body below the channel and its depletion region. In strong inversion (which is of interest here), when the channel exists, the surface voltage can be approximated well by $2\Phi_F$, i.e. double the voltage, that denotes the difference $W_F - W_i = -q\Phi_F$ between the Fermi levels of the doped (W_F) and undoped (W_i) semiconductor [115, p. 10 ff, p. 155 ff]; q denotes the elementary charge. In the literature, this voltage and the surface voltage are often referred to as potentials. Therefore, they are denoted by Φ instead of V.

$$Q_C = -WLC'_{\text{ox}} \left(V_{GB} - V_{\text{FB}} - 2\Phi_F - V_{SB} - \gamma\sqrt{2\Phi_F + V_{SB}} \right) , \quad (19.11)$$

where $V_{GB} = V_{s1}$ and $V_{SB} = V_1(t_{2'}) = V_2(t_{2'})$.

In addition to the quantities already used/described, W and L denote the width and length of the transistor channel. V_{FB} is the flat band voltage of the transistor [115, S. 148 ff]. For the sake of brevity, we use the oxide capacitance $C_{\text{ox}} = WLC'_{\text{ox}}$. Solving (19.10) for the common voltage of nodes (1) and (2) yields:

$$\begin{aligned} V_1(t_{2'}) &= V_2(t_{2'}) = V_{1,2}(t_{2'}) \\ &= \frac{V_1(t_1)\big(C_1 + C_{g1}\big) + V_2(t_1)\big(C_2 + C_{g2}\big) + \int_{t_1}^{t_{2'}} I(t)\,dt}{C_1 + C_2 + C_{g1} + C_{g2}}(1+m) + \Delta V_{\text{on}} \ . \end{aligned} \quad (19.12)$$

The results deviates from that of an ideal switch by the gain factor

$$m = \frac{C_{\text{ox}}}{C_1 + C_2 + C_{g1} + C_{g2}} \quad (19.13)$$

and by an error voltage caused by charge injection at turn-on

$$\begin{aligned} \Delta V_{\text{on}} = &\frac{(V_{s1} - V_{s0})(C_{g1} + C_{g2}) + C_{\text{ox}}(V_{s1} + V_{\text{FB}} - 2\Phi_F)}{C_\Sigma} \\ &+ \frac{C_{\text{ox}}^2\gamma^2/2}{C_\Sigma^2} + \sqrt{\left(\frac{Q_B C_\Sigma + C_{\text{ox}}^2\gamma^2/2}{C_\Sigma^2}\right)^2 - \frac{Q_B^2 - 2C_{\text{ox}}^2\gamma^2\Phi_F}{C_\Sigma^2}}. \end{aligned} \quad (19.14)$$

Q_B and C_Σ have been introduced as abbreviations for the parts which are independent from $V_{1,2}(t_{2'})$. They are

$$\begin{aligned} Q_B = &V_1(t_1)\big(C_1 + C_{g1}\big) + V_2(t_1)\big(C_2 + C_{g2}\big) + \big(V_{s1} - V_{s0}\big)\big(C_{g1} + C_{g2}\big) \\ &- \int_{t_1}^{t_{2'}} I(t)\,dt + C_{\text{ox}}\big(V_{s1} - V_{\text{FB}} - 2\Phi_F\big) \end{aligned} \quad (19.15)$$

and

$$C_\Sigma = C_{\text{ox}} + C_1 + C_2 + C_{g1} + C_{g2}. \quad (19.16)$$

As a rough approximation, it can be assumed, that in a practical application the charge injection is small compared to the intended changes of the charges on C_1 and C_2, and that it holds $C_{g1} \approx C_{g2} \ll C_1$ as well as $C_{g1} \approx C_{g2} \ll C_2$, and that the charge coming from the current source $\int_{t_1}^{t_{2'}} I(t)\,dt$ is negligible.

If the influence of the switching transistor was completely ignored for a moment ($C_{\text{ox}} \approx 0$, $Q_C \approx 0$), the voltage at time $t_{2'}$ could be calculated from simple charge equality:

$$V'_{1,2}(t_{2'}) = \frac{V_1 C_1 + V_2 C_2}{C_1 + C_2}. \tag{19.17}$$

The non-ideal switching transistor causes a deviation of that voltage. Because the deviation is small in practical cases (that is, what the circuit is dimensioned for) the channel charge and the charge going into the parasitic capacitances of the transistor can be obtained from the ideal $V'_{1,2}(t_{2'})$:

$$Q_C \approx Q'_C = -WLC'_{\text{ox}} \left(V_{s1} - V_{\text{FB}} - 2\Phi_F - V'_{1,2}(t_{2'}) - \gamma \sqrt{2\Phi_F + V'_{1,2}(t_{2'})} \right) \tag{19.18}$$

Overall, the deviation from an ideal switch, at turn-on, is approximately

$$\Delta V_{\text{on}} \approx \frac{Q'_C + (V_{s1} - V_{s0})(C_{g1} + C_{g2})}{C_1 + C_2}. \tag{19.19}$$

Until the switch is turned off again at time t_3, the voltages V_1 and V_2 are equal and take the value $V_{1,2}(t_3)$. After the turn-off V_2 is supposed to stay at that value $V_2(t > t_4) \approx V_{1,2}(t_3)$.

The charge coming from the current source during turn off

$$Q_{34} = \int_{t_3}^{t_4} I(t')\, dt' \tag{19.20}$$

also contributed to the deviation at turn-off. However, in practical applications, the time evolution of I and the duration $t_4 - t_3$ of the turn-off procedure are chosen, such that this contribution is negligible.

However, the charge injection cannot be neglected. The channel charge Q_C is distributed among the nodes (1) and (2) and causes a voltage deviation. Charging of the parasitic capacitances of the transistor contributes to the voltage deviation as well.

When the switching procedure is done very quickly ($t_4 - t_3 \approx 0$), the channel charge is approximately evenly distributed onto nodes (1) and (2), and the other capacitances at node (1) do not influence the charge injection to node (2). Charge equality

$$U_2(t_3)\big(C_2 + C_{g2}\big) - V_{s1} C_{g2} + \frac{Q_C}{2} = V_2(t_4)\big(C_2 + C_{g2}\big) - V_{s0} C_{g2} \tag{19.21}$$

yields the voltage at node (2) after turn-off

$$V_2(t_4) = V_2(t_3) + \Delta V_{\text{off}}\ . \tag{19.22}$$

It deviates from the ideal value $V_2(t_4) = V_2(t_3)$ by

$$\Delta V_{\text{off}} = \frac{\big(V_{s0} - V_{s1}\big) C_{g2} + Q_C/2}{C_2 + C_{g2}}. \tag{19.23}$$

The channel charge Q_C can again be obtained from (19.11) with $V_{GB} = V_{s1}$ and $V_{SB} = V_2(t_3) = V_1(t_3)$.

When the turn-off procedure is done slowly (however, fast enough such that Q_{34} from (19.20) is still negligible), the channel keeps conducting until V_s drops below the threshold voltage V_t. While the channel is still conducting, the channel charge is distributed according to the ratio of capacitances connected to nodes (1) and (2). At time t_4' , with $t_3 < t_4' < t_4$, the gate-to-source voltage $V_{GS} = V_s - V_1 = V_s - V_2$ drops below the threshold voltage $V - t$. For an exact calculation of t_4' and $V_t(t_4')$, one would need to consider that $V_2(t_4') \neq V_2(t_3)$ and consequently $V_t(t_4') \neq V_t(t_3)$. However, in practical applications the voltage disruption by charge injection is small compared to the signal range of V_2. Therefore, it can be assumed that the threshold voltage is not changing: $V_t(t_4') = V_t(t_3)$. It can be obtained from (19.7) with $V_{SB} = V_2(t_3) = V_1(t_3)$:

$$V_t = V_{t0} + \gamma \left(\sqrt{2\Phi_F + V_{1,2}(t_3)} - \sqrt{2\Phi_F} \right) . \tag{19.24}$$

U_s keeps dropping for $t > t_4'$. Now, only charge injection through C_{g2} contributes to further voltage change at node (2).

In both parts of the turn-off procedure, before and after t_4' , the voltage change can be obtained from charge conservation. For $t \leq t_4'$ it is

$$\begin{aligned} &V_{1,2}(t_3)\big(C_1 + C_2 + C_{g1} + C_{g2}\big) - V_{s1}\big(C_{g1} + C_{g2}\big) + Q_C \\ &\quad = V_{1,2}(t_4')\big(C_1 + C_2 + C_{g1} + C_{g2}\big) - \big(V_{1,2}(t_4') + V_t(t_3)\big)\big(C_{g1} + C_{g2}\big) , \end{aligned} \tag{19.25}$$

while for $t > t_4'$ it is

$$V_2(t_4')\big(C_2{+}C_{g2}\big) - \big(V_{1,2}(t_4'){+}V_t(t_3)\big)C_{g2} = V_2(t_4)\big(C_2{+}C_{g2}\big) - V_{s0}C_{g2} . \tag{19.26}$$

Solving for $V_2 t_4$ yields:

$$V_2(t_4) = V_{1,2}(t_3)\big(1 + m\big) + \Delta V_{\text{off}} . \tag{19.27}$$

The result deviates from the ideal value $V_{1,2}$ by the gain factor

$$m = \frac{C_1 + C_2 + C_{g1} + C_{g2}}{C_1 + C_2} \frac{C_2}{C_2 + C_{g2}} - 1 \tag{19.28}$$

and by the error voltage

$$\begin{aligned} \Delta V_{\text{off}} = \frac{C_2}{C_2 + C_{g2}} &\frac{\big(V_t(t_3) - V_{s1}\big)\big(C_{g1} + C_{g2}\big) + Q_C}{C_1 + C_2} \\ &+ \frac{C_{g2}}{C_2 + C_{g2}}\big(V_{s0} - V_t(t_3)\big). \end{aligned} \tag{19.29}$$

For further simplification, in many cases C_{g2} can be neglected compared to C_2.

Matching

Equally designed transistors, placed side by side, have differences in their electrical behaviour. This effect is known as mismatch. This has a number of reasons. Physical manufacturing processes, which determine, for example, layer thicknesses and doping profiles, are not exactly equal at different locations on the same wafer. They also deviate from wafer to wafer. Such deviations cause differences between identical circuit elements in different instances of the chip. This is referred to as global mismatch. Also, identical circuit elements placed at a long distance on the same chip deviate from each other. This is attributed to global mismatch as well.

A number of circuit design techniques is known to sufficiently suppress the effects of global mismatch, so e.g. the operating point and gain factor are only little sensitive to deviations of the electrical properties of the circuit elements. These techniques have been applied in the design and are not explicitly mentioned.

19.4 Optical Generation

Stefan Henker

Irradiance E describes the amount of radiant flux impinging on a unit surface area. By referencing irradiance with wave length λ, the spectral irradiance E^λ with unit $Wm^{-2}nm^{-1}$ is defined. >Thereof, the spectral photon flux density $\mathit{\Pi}^\lambda$ as $m^{-2}s^{-1}nm^{-1}$ is derived as the number of photons per area, time and wave length. The correlation between spectral irradiance E^λ and photon flux density $\mathit{\Pi}^\lambda$ is given by:

$$\mathit{\Pi}^\lambda = \frac{1}{h \cdot f} \cdot E^\lambda = \frac{\lambda}{h \cdot c} \cdot E^\lambda \ , \tag{19.30}$$

with Planck's constant h, speed of light c and it's frequency f. When electromagnetic radiation is impinging on a solid state body, the radiation is partially reflected and partially absorbed. The remaining radiation is transmitted. The absorption of the photon flux on a position x in the solid state body is proportional to the photon flux density $\mathit{\Pi}^\lambda$ at this position. The photon flux density over wave length shall be defined as $\pi(\lambda)$, and the position dependent photon flux density as $\pi(\lambda, x)$.

Absorption in a solid state body is described by the Lambert-Beer law. In the one dimensional case, where the radiation is impinging vertically on the surface, the following equation holds:

$$\pi(\lambda, x) = \pi(\lambda, 0) \cdot e^{-\beta(\lambda) \cdot x} \ . \tag{19.31}$$

The term $\pi(\lambda, 0)$ is the photon flux density at the surface of the solid state body, reduced by the reflected part. The photon absorption inside the solid state body is a function of the wave length. This is described by the material dependent absorption coefficient $\beta(\lambda)$ (Figure 19.7). The distance $\beta(\lambda)^{-1}$ is called the penetration depth, as at $x = \beta(\lambda)^{-1}$ the photon flux density for wave length λ is reduced by the factor e^{-1}. In general, the penetration depth increases with the wave length of the light. Photons with high energy (short wave length) are absorbed near the surface, while photons of longer wave length impinge deeper into the solid state body. Thus, low energy photons are absorbed in a larger volume.

Further, absorption is strongly dependent on the radiated material. Metals absorb photons of a large range of wave lengths, since they contain scores of electrons in the conduction band. These electrons can be transferred to almost any higher energy level by interaction with photons, and to some extent even leave the solid state body. In contrast, semiconductors contain only few free charge carrier. The main mechanism of photon absorption here is photo generation, the generation of electron-hole pairs. In order to generate a free charge carrier inside the intrinsic semiconductor, the impinging photon needs an energy $h \cdot f$ larger than the material dependent band gap ΔW. Otherwise the photon is transmitted through the semiconductor without interaction. In doped semiconductors there is another possible way to generate photos , where electrons transcend the lower gap between the conduction band and the donor level. However, in all cases there is a maximum wave length, that contributes to photo generation:

$$\lambda_{max} = \frac{h \cdot c}{\Delta W} \ . \tag{19.32}$$

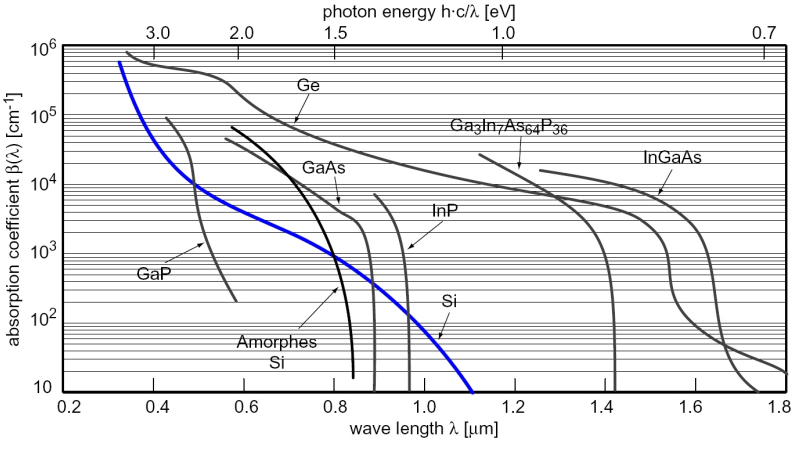


Fig. 19.7 Absorption coefficients over wave length for different materials

However, not every absorbed photon results in a usable electron-hole pair. For instance, photons with high energy are absorbed near the surface and recombine almost immediately because of surface conditions that act as recombination centers. Thus, there is also a lower limit of wave lengths, that contribute to photo generation.Additionally, semiconductor material can also absorb photons of lower energy than the band gap. But these do not contribute to photo generation, as, similar to photon absorption in metals, conduction band electrons are raised to a higher energy level. When these electrons return to their natural level, the energy is transferred to thermal energy in the form of lattice vibration. The number of generated charge pairs with respect to the number of absorbed photons is called spectral quantum efficiency $\eta_q(\lambda)$. The spectral optical generation rate $G(\lambda, x)$ results from differentiation of the spectral photon flux density over x with respect to the quantum efficiency as:

$$G(\lambda, x) = -\eta_q(\lambda) \cdot \frac{\delta}{\delta x}\pi(\lambda, x) = \eta_q(\lambda) \cdot \pi(\lambda, 0) \cdot \beta(\lambda) \cdot e^{-\beta(\lambda)\cdot x} \ . \qquad (19.33)$$

This can be achieved by applying an external voltage or by an intrinsic space charge region.

Thus, photo generation depends on material coefficients, the depth x in the semiconductor, the photon flux density, and their wave length. In the case of crystalline silicon, photo generation occurs for wave lengths of $300\,nm$ to $1200\,nm$. But in semiconductors, the number of free charge carriers is in equilibrium; charge recombination is counteracting the photo generation. The recombination rate is proportional to the number of free charges. Photogenerated charges in an irradiated semiconductor recombine and emit energy in the form of radiation or lattice vibration, depending on the specific recombination mechanism. Thus, a measurement of light intensity is not possible using photo generation alone. The technical usage of photo generation, i.e. photo voltaic power generation or radiation detection, needs to be able to separate the generated electron-hole pair in order to prevent their recombination.One method used to separate charges is an electric field. This can be achieved by applying an external voltage or by creating a space charge region.

References

[1] Cronin, J.: Mathematical aspects of Hodgkin-Huxley neural theory. Cambridge University Press, New York (1987)

[2] Jones, J.P., Palmer, L.A.: An evaluation of the two-dimensional gabor filter model of receptive fields in cat striate cortex. Journal of Neurophysiology 58(6), 1233–1258 (1987)

[3] Jones, J.P., Stepnoski, A., Palmer, L.A.: The two-dimensional spectral structure of simple receptive fields in cat striate cortex. Journal of Neurophysiology 58(6), 1212–1232 (1987)

[4] Jones, J.P., Palmer, L.A.: The two-dimensional spatial structure of simple receptive fields in cat striate cortex. Journal of Neurophysiology 58(6), 1187–1211 (1987)

[5] von der Malsburg, C.: Zelle, Gehirn, Computer, und was sie sich zu erzählen haben. Festvortrag zur Verleihung der Ehrenbürgerwürde der Ruhr-Universität Bochum an Manfred Eigen (2001)

[6] von der Malsburg, C.: Self-organisation in the brain. In: Arbib, M. (ed.) The Handbook of Brain Theory and Neural Networks, 2nd edn., pp. 1002–1005. MIT Press, Cambridge (2002)

[7] Baddeley, R., Abbott, L.F., Booth, M.C., Sengpiel, F., Freeman, T., Wakeman, E.A., Rolls, E.T.: Responses of neurons in primary and inferior temporal visual cortices to natural scenes. In: Proc. R. Soc. Lond. B,264, pp. 1775–1783 (1997)

[8] Triesch, J.: Synergies between intrinsic and synaptic plasticity mechanisms. Neur. Comp. 19, 885–909 (2007)

[9] Laughlin, S.B., Sejnowski, T.J.: Communication in neuronal networks. Science 301, 1870–1874 (2003)

[10] Hodgkin, A.L., Huxley, A.F.: A quantitative description of ion currents and its applications to conduction and excitation in nerve membranes. Journal of Physiology 117(4), 500–544 (1952)

[11] McCulloch, W.S., Pitts, W.: A logical calculus of ideas immanent in nervous activity. Bull. Math. Biophys. 5, 115–133 (1943)

[12] Steinbuch, K.: Die Lernmatrix. Kybernetik 1(1), 36–45 (1961)

[13] Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. PNAS 79, 2554–2558 (1982)
[14] Palm, G.: On associative memory. Biological Cybernetics 36, 19–31 (1980)
[15] Rosenblatt, F.: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington (1961)
[16] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. In: Rumelhart, D.E., McClelland, J.L. (eds.) Parallel Distributed Processing: Explorations in the Microstructures of Cognition, Vol. I, pp. 318–362. MIT Press, Cambridge (1986)
[17] Bishop, C.: Neural Networks for Pattern Recognition. Oxford: University Press, Oxford (1995)
[18] Zell, A.: Simulation neuronaler Netze. Addison-Wesley, Reading (1994)
[19] Haykin, S.: Neural Networks. Macmillan, Basingstoke (1994)
[20] Geman, S., Bienenstock, E., Doursat, R.: Neural networks and the bias/variance dilemma. Neural Computation 4, 1–58 (1992)
[21] von der Malsburg, C.: The correlation theory of brain function. In: Domany, E., van Hemmen, J.L., Schulten, K. (eds.) Models of Neural networks II, chapter 2, pp. 95–119. Springer, Heidelberg (1994)
[22] Müller, B.: Sturm auf das Gehirn. In: Bild der Wissenschaft, Feb. 1993, pp. 102–104 (1993)
[23] Deco, G., Schürmann, B.: Information Dynamics: Foundations and Applications. Springer, Heidelberg (2001)
[24] Gerstner, W., Kistler, W.: Spiking Neuron Models. Cambridge University Press, Cambridge (2002)
[25] Singer, W.: Binding by synchrony. Scholarpedia 2(12), 1657 (2007)
[26] Phillips, P.J., Scruggs, W.T., O'Toole, A.J., Flynn, P.J., Bowyer, K.W., Schott, C.L., Sharpe, M.: Frvt 2006 and ice 2006 large-scale results (March 2007)
[27] Daugman, J.: Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. IEEE Transactions on Acoustics, Speech and Signal Processing 7, 1169–1179 (1988)
[28] Fellemann, D.J., Van Essen, D.C.: Distributed hierarchical processing in the primate cerebral cortex (Overview over anatomical knowledge of visual cortex). Cerebral Cortex 1, 1 (1991)
[29] Rolls, E.T.: Neurophysiological mechanisms underlying face processing within and beyond the temporal cortical visual areas. Philosophical Transactions of the Royal Society of London, Series B: Biological Sciences 335, 11–21 (1992)
[30] Singer, W.: Neuronal synchrony: A versatile code for the definition of relations? Neuron 24, 49–65 (1999)
[31] Shadlen, M.N., Movshon, J.A.: Synchrony unbound: A critical evalutation of the temporal binding hypothesis. Neuron 24, 67–77 (1999)
[32] von der Malsburg, C., Schneider, W.: A neural cocktail-party processor. Biological Cybernetics 54, 29–40 (1986)

[33] Sporns, O., Tononi, G., Edelman, G.M.: Modeling perceptual grouping and figure-ground segregation by means of active re-entrant connections. PNAS 88, 129–133 (1991)
[34] von der Malsburg, C., Buhmann, J.: Sensory segmentation with coupled neural oscillators. Biological Cybernetics 67, 233–242 (1992)
[35] Wang, D., Buhmann, J., von der Malsburg, C.: Pattern segmentation in associative memory. Neural Computation 2, 94–106 (1990)
[36] Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by a shift in position. Biological Cybernetics 36, 193–202 (1980)
[37] Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust object recognition with cortex-like mechanisms. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(3), 411–426 (2007)
[38] Koch, C.: Biophysics of Computation. Sinauer Associates (2001)
[39] Nicholls, J.G., Martin, A.R., Wallace, B.G., Fuchs, P.A.: From Neuron to Brain: A Cellular and Molecular Approach to the Function of the Nervous System, 4th edn. Oxford University Press, Oxford (1999)
[40] Johnston, D., Wu, S.M.-S.: Foundations of Cellular Neurophysiology. MIT Press, Cambridge (1994)
[41] Heittmann, A., Ramacher, U.: Correlation-based feature detection using pulsed neural networks. In: IEEE 13th Workshop on Neural Networks for Signal Processing, NNSP, pp. 479–488 (2003)
[42] Heittmann, A., Ramacher, U.: Pulsed neural networks for feature detection using dynamic synapses. In: Fourth International ICSC Symposium on Engineering of Intelligen Systems, EIS (2004)
[43] Landau, L.D., Lifshitz, L.M.: Mechanics. Elsevier, Amsterdam (1975)
[44] Dayan, P., Abbott, L.F.: Theoretical Neuroscience. MIT Press, Cambridge (2001)
[45] Maffei, L., Fiorentini, A.: The unresponsive regions of visual cortical receptive fields. Vision Res. 16, 1131–1139 (1976)
[46] Lücke, J.: Information Processing and Learning in Networks of Cortical Columns (Dissertation). Shaker Verlag (2005)
[47] Thorpe, S.: Identification of rapidly presented images by the human visual system. Perception 17(A77), 415 (1988)
[48] Bi, G., Poo, M.: Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. Journal of Neuroscience 18, 1464–1472 (1998)
[49] Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimisation. IEEE Transactions on Evolutionary Computation 1(1), 67–82 (1997)
[50] Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation 3, 79–87 (1991)
[51] Carpenter, G.A., Grossberg, S.: The art of adaptive pattern recognition by a self-organizing neural network. Computer 21(3), 77–88 (1988)

[52] Desimone, R., Duncan, J.: Neural mechanisms of selective visual attention. Annual Review of Neuroscience 18, 193–222 (1995)
[53] Vidal, J.R., Chaumon, M., O'Regan, J.K., Tallon-Baudry, C.: Visual grouping and the focusing of attention induce gamma-band oscillations at different frequencies in human magnetoencephalogram signals. J. Cognitive Neuroscience 18(11), 1850–1862 (2006)
[54] Feldman, J.A.: Dynamic connections in neural networks. Biol. Cybern. 46, 27–39 (1982)
[55] von der Malsburg, C.: Dynamic link architecture. In: Arbib, M. (ed.) The Handbook of Brain Theory and Neural Networks, 2nd edn., pp. 365–368. MIT Press, Cambridge (2002)
[56] Olshausen, B., Anderson, C., van Essen, C.: A eurobiological model of visual attention and invariant recognition based on dynamic routing of information. The Journal of Neuroscience 13(11), 4700–4719 (1993)
[57] Jitsev, J., Sato, Y.D., von der Malsburg, C.: A neural system for scale and orientation invariant correspondence finding. In: Proc. COSYNE (2008)
[58] Wolfrum, P., Wolff, C., Lücke, J., von der Malsburg, C.: A recurrent dynamic model for correspondence-based face recognition. In: Journal of Vision (in press)
[59] Wiskott, L., von der Malsburg, C.: Face recognition by dynamic link matching. In: Sirosh, J., Miikkulainen, R., Choe, Y. (eds.) Lateral Interactions in the Cortex: Structure and Function, chapter 4. The UTCS Neural Networks Research Group, Austin, ISBN 0-9647060-0-8 (1996)
[60] Bergmann, U., von der Malsburg, C.: Ontogenesis of invariance transformations. In: Computational and Systems Neuroscience, Cosyne (2008)
[61] Biederman, I., Kalocsai, P.: Neurocomputational bases of object and face recognition. Phil. Trans. Roy. Soc.: Biological Sciences 352, 1203–1219 (1997)
[62] Biederman, I.: Recognition-by-components: A theory of human image understanding. Psychological Review 94, 115–147 (1987)
[63] Gibson, J.J.: The Perception of the Visual World. The Riverside Press, Cambridge (1950)
[64] Ungerleider, L.G., Mishkin, M.: Two cortical visual systems. In: Analysis of Visual Behavior, pp. 549–586. MIT Press, Cambridge (1982)
[65] Gerstner, W., Kistler, W.: Spiking Neuron Models: An Introduction. Cambridge University Press, New York (2002)
[66] Koch, C.: Biophysics of Computation: Information Processing in Single Neurons (Computational Neuroscience). Oxford University Press, Oxford (2004)
[67] Heittmann, A., Matolin, D., Schreiter, J., Schüffny, R., Ramacher, U.: An analog vlsi pulsed neural network for image segmentation using adaptive connection weights. In: International Conference on Artificial Neural Networks (2002)

[68] Schreiter, J., Ramacher, U., Heittmann, A., Matolin, D., Schüffny, R.: Analog implementation for networks of integrate-and-fire neurons with adaptive local connectivity. In: IEEE International Workshop on Neural Networks for Signal Processing, Martigny, Switzerland, pp. 657–666 (2002)

[69] Schreiter, J., Döge, J., Matolin, D., Schüffny, R., Heittmann, A., Ramacher, U.: Image segmentation by a pcnn with adaptive weights. In: Visualization, Imaging, and Image Processing - VIIP 2003, Benalmádena, Spain (2003)

[70] Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural Computation 14(11), 2531–2560 (2002)

[71] Shepherd, G.M.: The Synaptic Organization of the Brain, 4th edn. Oxford University Press, New York (2004)

[72] Eckhorn, R.: Neural mechanisms of scene segmentation: recordings from the visual cortex suggest basic circuits for linking field models. IEEE Transactions on Neural Networks 10(3), 464–479 (1999)

[73] Freeman, T.C.B., Durand, S., Kiper, D.C., Carandini, M.: Suppression without inhibition in visual cortex. Neuron 35(4), 759–771 (2002)

[74] Vittoz, E.A.: Handbook of Neural Computation. Oxford University Press, Oxford (1997)

[75] Möschwitzer, A.: Grundlagen der Halbleiter- und Mikroelektronik. Carl Hanser Verlag (1992)

[76] Chouikha, M.B., Lu, G.N., Sedjil, M., G., Sou, A.G.: Buried triple p-n junction structure in a bicmos technology for color detection. In: Proc. of the Bipolar/BiCMOS Circuits and Technology Meeting, Minneapolis, MN, USA, September 1997, pp. 108–111 (1997)

[77] Herzog, P.G., Knipp, D., Stiebig, H., König, F.: Characterization of novel three and six channel color moire free sensors. SPIE 3648, 48–59 (1999)

[78] Wyszecki, W.S., Stiles, G.: Color Science: Concepts and Methods, Quantitative Data and Formula, 2nd edn. John Wiley & Sons, New York (1982)

[79] Doswald, Haflinger, Blessing, Felber, Niederer, Fichtner: A 30-frames/s megapixel real-time cmos image processor. IEEE Journal on Solid-State Ciruits 35(11), 1732–1743 (2000)

[80] Ramacher, U., Koren, I., Geib, H., Kirmser, S., Heer, C., Schlüßler, J.-U., Dohndorf, J., Werner, J.: Single-chip video camera with multiple integrated functions. In: Solid-State Circuits Conference ISSCC'99, pp. 306–307 (1999)

[81] Huppertz, J.: 2-D CMOS Bildsensorik mit integrierter Signalverarbeitung. PhD thesis, Gerhard-Mercator-Universität - Gesamthochschule Duisburg (2000)

[82] Fossum, E.R.: Cmos image sensors: Electronic camera-on-chip. IEEE Transactions on Electron Devices 44, 1689–1698 (1997)

[83] Rullmann, M.: Entwurf und Implementierung von Algorithmen für die Korrektur von Bildfehlern bei CMOS-Sensoren. Master's thesis, TU Dresden (January 2003)
[84] Fossum, E.R.: Active pixel sensors (aps) - are ccds dinosaurs? Proceedings SPIE 1900, 214 (1992)
[85] Nakamura, J., Pain, B., Nomoto, T., Nakamura, T., Fossum, E.R.: On-focal-plane signal processing for current-mode active pixel sensors. IEEE Transactions on Electron Devices 44, 1747–1758 (1997)
[86] Henson, W.K., Yang, N., Kubicek, S., Vogel, E.M., Wortman, J.J., De Mayer, K., Naem, A.: Analysis of leakage currents and impact on off-state power consumption for cmos technology in 100-nm regime. IEEE Transactions on Electron Devices 47(7), 1393–1400 (2000)
[87] Henker, S., Mayr, C., Schlüß, J.-U.: ler, R. Schüffny, U. Ramacher, and A. Heittmann. Active Pixel Sensor Arrays in 90/65nm CMOS-Technologies with vertically stacked photodiodes . In: International Image Sensor Workshop, IISW 2007, Ogunquit, USA (2007)
[88] IBIS4-6600 High resolution 6.6 M Pixel Rolling shutter CMOS Image sensor Datasheet. San Jose, CA (2005)
[89] Henker, S., Graupner, A., Getzlaff, S., Schreiter, J., Schüffny, R.: A new hierarchical simulator for highly parallel analog processor arrays. In: Merker, R., Schwarz, W. (eds.) System Design Automation - Fundamentals, Principles, Methods, Examples, pp. 176–182. Kluwer Academic Publishers, Dordrecht (2000)
[90] Liu, W., et al.: BSIM3v3.3.0 MOSFET Model- Users' Manual. UC Berkeley (July 2005)
[91] Kundert, K.S.: The Designer's Guide to SPICE and SPECTRE. Kluwer Academic Publishers, Dordrecht (1995)
[92] Enz, C., Krummenacher, F., Vittoz, E.A.: An analytical MOS transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications. Analog Integrated Circuits and Signal Processing 8(1), 83–114 (1997)
[93] Petzold, L.R., Brown, P.N., Hindmarsh, A.C., Ulrich, C.W.: DASKR. Dokumentierte Software
[94] Hindmarsh, A.C., Brown, P.N., Grant, K.E., Lee, S.L., Serban, R., Shumaker, D.E., Woodward, C.S.: Sundials: Suite of nonlinear and differential/algebraic equation solvers. ACM Transactions on Mathematical Software 3(31), 363–396 (2005)
[95] Ottmann, T., Widmayer, P.: Algorithmen und Datenstrukturen, vol. 4. Spectrum Akademie Verlag, Heidelberg, Berlin (2002)
[96] Mayr, C., Eisenreich, H., Schreiter, J., Schüffny, R.: Pulse capture and distribution for neural pulse-based computations. In: Dresdner Arbeitstagung Schaltungs- und Systementwurf 2006, pp. 85–90 (2006)
[97] Van Rullen, R., Thorpe, S.J.: Rate coding versus temporal order coding: What the retinal ganglion cells tell the visual cortex. Neural Computation 13(6), 1255–1283 (2001)

[98] Chipperfield, A., Fleming, P., Pohlheim, H., Fonseca, C.: Genetic algorithm toolbox for use with matlab (1996)
[99] Riesenhuber, M., Poggio, T.: Hierarchical models of object recognition in cortex. Nature Neuroscience 2(12), 1019–1025 (1999)
[100] ITRS. International Technology Readmap for Semiconductors (2007)
[101] Elmore, W.C.: The transient analysis of damped linear networks with particular regard to wideband amplifiers. J. Applied Physics 19(1), 787–790 (1948)
[102] Kaiser, A.: Dreidimensionale Systemintegration: Technologische Entwicklung und Anwendung. PhD thesis, Cuvillier Verlag, University of Ulm (2007)
[103] Bschorr, M., Pfleiderer, H.-J., Benkart, P., Kaiser, A., Munding, A., Kohn, E., Heittmann, A., Hübner, H., Ramacher, U.: Eine Test- und Ansteuerschaltung für eine neuartige 3D Verbindungstechnologie. Advances in Radio Science 3, 305–310 (2005)
[104] Payne, C.C.: Silica sol composition for polishing silicon wafers. U.S. Patent No.4462188 (1984)
[105] Mandle, R.M.: Process for preparing a polishing compound and product. U.S. Patent No.3298807 (1967)
[106] Lärmer, F., Schilp, A.: Method of anisotropically etching silicon. U.S. Patent No.5501893, DE4241045, EP625285 (1992)
[107] Park, Y.-B., Kang, J.-K., Rhee, S.-W.: Effect of N_2O/SiH_4 ratio on the properties of low-temperature silicon oxide films from remote plasma chemical vapour deposition. Thin Solid Films 280, 43 (1996)
[108] Stein, J.: Niedertemperatur-Deposition von Siliziumdioxid mittels Remote-PECVD. PhD thesis, RWTH Aachen (1999)
[109] Itsumi, M.: SiO2 in Si Microdevices. Springer, Heidelberg (2002)
[110] Frenkel, J.: On pre-breakdown phenomena in insulators and electronic semiconductors. Physical Reviews 43, 647–648 (1938)
[111] Beynon, J.: The electrical properties of anodically grown silicon dioxide films. Solid State Electronics 16, 309–314 (1973)
[112] Massalski, T.B., Murray, J.L., Bennet, L.H., Baker, H.: Binary Alloy Phase Diagrams. American Society for Metals, Metals Park (1986)
[113] Bartels, F., Morris Jr., J.W.: Intermetallic phase formation in thin solid-liquid diffusion couples. Journal of Electronic Materials 23, 787–790 (1994)
[114] Hübner, H., Ehrmann, O., Eigner, M., Gruber, W., Merkel, R., Ramm, P., Roth, M., Weber, J., Wieland, R.: Face-to-face chip integration with full metal interface. In: Proceedings Advanced Metallization Conference AMC 2002, pp. 53–58 (2002)
[115] Hoffmann, K.: From Transistor Design to Large Scale Integrated Circuits. John Wiley & Sons, New York (2004)

Index

A

A-before-B detector 308, 310
action potential 186–189, 237
active pixel sensors (APS) 217
AER
  address event representation 251
  FIFO (first-in-first-out memory) 261–263
AER chip 300
  segments 301
  signalling cell 300
AER coding 250
AER coding unit 259–262
AER unit 258–259
  arbiter 263
anti-coincidence detector 99, 137, 147
artificial vision system 299
axon 12, 107, 186, 263, 335

B

basic circuits (CMOS)
  basic transistor circuits 338–344
  current mirror 200
  current source 197
  current storage cell 231
  differential amplifier 200
  differential pair 200
  switches 338–343
  two-quadrant multiplier 208
  weight storage 207
binding 104
  object binding 2
  of features 136–138, 148
biological neurons 186
  bipolar cells 107, 150
  ganglion cells 107, 150, 151
  horizontal cells 107
biological synapses 186–191
  neurotransmitter 187
  synaptic cleft 187
bipolar cells 107, 150

C

CAM cell 316
cell membrane 186, 187
characteristic functions 50
CMOS 185, 214, 228, 269, 299
  mismatch parameters 199
  structures for photodiodes 215
coincidence detector 99
compartmental model 189
cones 149
content addressable memory (CAM) 315
correlation 2–3, 14, 43, 79, 83–88, 100–103, 120, 136, 175, 194, 255

D

dark current 212, 222
decorrelation 3, 85, 255
dendrite 93–99, 126, 186, 263
  dendritic equation 99
  dendritic synapse 99, 127, 137
dendritic equation 99, 127

dendritic synapse
  differential equation 93, 127
dentritic tree 189
detector profile 111–115, 157
  connection density 114–117
difference counter 183, 256
distance invariance 148

E

emerging pattern 90
ensemble 44
entropy 2, 4, 31, 47–58, 60–81, 83, 86, 104, 120, 175
  signal entropy 53
EPSC (excitatory postsynaptic current) 187
EPSP (excitatory postsynaptic potential) 188
event driven simulator 328

F

feature cascade 5, 149–160, 175, 185–186, 267, 322
  resolution pyramid 149–160
feature detection 14–18, 105, 148, 160, 185, 255–267
  neurobiological 264
feature detector 105–117, 121–124, 148, 183, 230, 247–248, 255–264, 317
  implementation 157
feature recogniser 126, 160, 247–254, 317, 322
  NPU architecture 317–319
feature recognition 2, 18, 119, 123
filter 112
  filter profile 115
first generation vision cube 299–323
fixed pattern noise 216
fovea centralis 150
frequency distribution 28
functional derivative 65

G

Gabor wavelet 16–19, 107–117, 119–136, 142, 161, 183, 255, 263–264, 269
  inverse 162–164
gabor wavelets
  detector profiles 263
ganglion cells 107, 150
ganglion geniculatum laterale 107
gradient detector 105–106, 113, 157, 159
  „2-before-1“ detector 260
  A-before-B detector 308, 310
  pulse difference circuit 263

H

Hamilton-Jacobi equation 51
Hamiltonian 51
head concluder 148
head detection 141–147
head detector 142–147, 170
head recogniser 148
Hodgkin-Huxley model 189, 335–337
horizontal cells 107
hyperpolarisation 107

I

IAF neuron
  circuit concept 195–205
  layout 205
  small signal analysis 203
implementation (CMOS)
  adaptive synapses 205–210
  analogue image storage 230–231
  charge current source 197–199
  current mode circuit technology 249
  DA-converters 230–231
  feature detector 257–267
  feature recogniser 252–254
  multichannel photosensors 222–226
  neuron 195–205
  photosensors 210–230
  pixel cells 222–230
  pulsing pixel cell 227–230
  threshold switch 199–201
information processing 2
information-carrying transfer function 33
inner screen 161–170
integrate-and-fire model (IAF) 190–194
invariance 122

distance-invariant feature pyramid 149
independence from deformations 125
independence from location 124
invariance to change of illumination 122
invariance to distance 122
invariance to rotation 122
scale invariant object recognition 152
ion channel 186, 189, 335
ion channels 335
ion pumps 186
ionotrop receptor 187
IPSC (inhibitory postsynaptic current) 187
IPSP (inhibitory postsynaptic potential) 188
ISI histogram 26

L

leaky integrator 190
LGN 107, 108

M

macula lutea 150
membrane conductance 188
metabotrop receptor 187
mismatch parameters 199

N

nervus opticus 107
net
cold net 30
hot net 29
nets for
feature detection 105
feature recognition 119
robust head detection 141
NPU
neural processing unit 256

O

object binding 2
object detector 183
object recogniser 154
object recognition 138, 184
object components 121
principles of object recognition 121
operating range 33
optical axis 150
optical nerve 150
orientation detector 108

P

partial entropy 68
partition 2, 47, 59–80, 83, 84, 102, 103
pattern recognition
spacial 38
perception of colour 149
photo generation 346
photodiode 211–213
generation rate 212
spectral sensitivity 212–216, 222–226
photoelectric effect 210
photopic vision 150
photosensor
absorption coefficient 217
CCD 216
CMOS 216–231
colour filter 215–216
integrating photosensor 213
moat 222
multichannel photosensors 214–216
optical generation 344–346
pixel circuit (CMOS) 217–226
quantum efficiency 213, 217, 346
model to estimate 217
reflection coefficient 217
pixel cell 258
electronic shutter 218–222
pulsing pixel cell 227–230
read-out circuit 219–222
reset circuit 218–219
point model 189
postsynaptic depolarisation 188
primary visual cortex 108
PSC (postsynaptic current) 187
PSP (postsynaptic potential) 188, 191
pulse difference computation 260–261
2-before-1 detector 260, 263, 266
A-before-B detector 308, 310

gradient detector 105, 106, 117, 159, 263, 308, 309, 314, 315
symmetric computation of pulse difference 95
pulse difference counter 263
pulse patterns 83
pulse processing 45
pulse rate 33
mean pulse rate 34
pulse rate difference 98
pulse wave 88, 103, 185, 252–254
pulses
pulse differences 93–96
pulse sums 93–96

Q

quantum efficiency 213, 217, 346
model to estimate 217

R

receptive fields 108–117, 158, 194, 263–264
recognition architecture 159, 160
recognition chain 2, 21, 104, 142, 149, 185
resolution 35
resolution pyramid 149–160, 299–301
retina 17, 106–114, 123, 263
robust
feature recognition 17, 123
head detection 141
object detection 17
object recognition 138
robustness 44
rod cells 149

S

scotopic vision 150
segmentation 91, 104, 124
per synchronisation 91
segmentiser 254
selectivity 128
self-organisation 11–13, 173–181
semiconductive photosensors 211–230
shapes
arbitrary 106
curves 106, 115
edges 105, 106, 115
textures 106
shunting inhibion 188
signal energy 47, 48
membrane energy 48
synapse energy 48
total energy 48
variation of the signal energy 56
signal equations 21
simple cells 108
simulation of microelectronic neural systems 233–246
EKV DC model 234
implementation 246
modelling 234, 246
neuron model 234
simulator 233
synapse model 235
simulation time 47
SoLID (solid-liquid-interdiffusion soldering process) 292–294
spike response model (SRM) 189
spot detector 248, 249, 254
stacking technology (3D)
anodic oxidation 285
ASE process 279
assembly 292–298
attachment of auxiliary carrier 278
batch process 274
Bosch process 279
chemical mechanical polishing (CMP) 277
chip-to-wafer bonding 274
CMOS, preprocessing 274, 276
connection density 272, 273, 275, 276
deep reactive ion etching (DRIE) process 279
electrical breakdown 283
electrical properties of 3D interconnections 296–298
electroplating 288
etch markers 276
etching of vias 279–280
Fowler-Nordheim tunneling 283
fundamentals 271–273
grinding 278
high-precision mask alignment 276
insulation of vias 281–287

known-good-dies (KGD) 274
leakage current 282
low temperature budget 274
low temperature processes 273
LT-PECVD process 281
mechanical bonds 271
metallisation of Vias 287–291
parasitic capacitances 272
parasitic resistances 272
pinholes 285
Poole-Frenkel effect 283
power consumption 270, 271, 298
reduction of wiring length 271
residual wafer thickness 276
signal integrity 273
skew, jitter, crosstalk 273
solid-liquid interdiffusion soldering process (SoLID) 292–294
thinning of wafers 276–278
via filling 288
wafer-to-wafer bonding 274
synapses 17
constant 3, 17, 43–45, 47, 58, 59, 81, 88–104, 124, 138
dendritic equation 99, 127
dendritic synapse 94, 127, 185
differential equation 23, 48–51, 84–85, 93, 94, 127
dynamic 3, 17, 18, 47, 58, 83–104, 117, 119–139, 184, 254, 255, 313
excitatory 194
excitatory synapses 91
inhibitive synapse 91
inhibitory 194
synaptic equation 23, 51, 84–85, 127, 193
synaptic conductance 188
synaptic efficiency 189
synaptic equation 23, 51, 84–85, 127, 193
synchronisation 3, 18, 85, 185, 194
pulse wave 252
statistics 103
wave front 90
synchronisation detector 99, 100, 126–139, 183, 248, 322
system integration (3D) 270–298

T

threshold switch
hysteresis 195
time coding 100

V

V1 region 194
V4 region 194
vision cube 298–321
AER chip 300–304
CAM array 315
chip for feature detection 310–312
chip for feature recogniser 312–321
chip stack 322
crossbar 259
feature synchroniser 319–321
imager chip 299–300, 322
NPU Array 304–310
NPU quad 306
power consumption 185, 315, 322
routing device 259–267
stacked architecture (3D) 321
synchronisation detector 312
visual cortex 150, 194
visual system 107
overview 107

W

wave front 90
weak inversion 199

Y

yellow spot 150

Z

zone 121–139
zone architecture 148, 159, 185, 313, 319
zone detector 137–139, 142–147
zone detector 137–139, 142–147
architecture 137