DISS. ETH NO. 21347

Neuromorphic Implementation of a Saliency-based Visual Selective Attention System

A dissertation submitted to

ETH ZURICH

for the degree of

Doctor of Sciences

presented by

DANIEL EDUARD SONNLEITHNER

Dipl.-Ing., Universität Karlsruhe (TH)

23. December 1982

citizen of

Vully-les-Lacs, Switzerland

accepted on the recommendation of

Prof. Dr. Rodney Douglas, examiner

Prof. Dr. Giacomo Indiveri, co-examiner

Prof. Dr. Ernst Niebur, co-examiner

2013

TO INKEN MARIE.

Disclaimer

I hereby declare that the work in this thesis is that of the candidate alone, except where indicated in the text, and as described below.

The description of experiments and results of creating a scan-path, Sec. 4.2, was already published in Sonnleithner and Indiveri [2012].

The use of “we” in the thesis refers to the above-mentioned people in the relevant sections.

Abstract

Visual perception is one of our most important senses. To be able to see our eyes transform the light signal into spiking data streams. Our brain extracts from this data relevant information. To provide vision to a mobile robotic system both computational steps have to be implemented. With today's technology it is possible to build visual sensors with high resolution and high recording frame rates that dissipate little power. The sensors provide a clear, detailed view of the robot's environment. Therefore the first aspect of vision is implemented. The second aspect is to extract relevant information from the visual data. Information are relevant for the robot if they allow its interaction with the environment, e.g. determining its location or recognizing obstacles. Therefore, the extraction has to happen in real time.

The problem for the robot is to extract relevant information from the huge amount of visual data provided by its sensors in real time with limited computational resources.

In biology an analogy of this phenomenon can be observed: Our eyes provide far more data than the human brain can process. Nevertheless we are able to interact with our environment in real time. The mechanism that allows us to extract the relevant information from the data provided by our eyes in real time is called *selective attention* [Treisman and Gelade, 1980]: Only a subset of the visual data is processed in detail, the rest is discarded. A preprocessing system identifies regions in the visual space that are salient. The visual data from these regions is processed further by our brain in a serial fashion. An alternative to master the described problem is to adapt this bio-inspired strategy to robotics. In this thesis I present a neuromorphic multi-chip system that is derived from a saliency-based selective attention model [Koch and Ullman, 1985]. My proposed solution uses building blocks derived from the brain: emulates of neurons and synapses. Therefore, it achieves very efficient computations. To estimate the most relevant regions of the input scene the model uses a "saliency map": this map assigns to each pixel of the input image a value for its saliency. In the model, saliency is computed by using center-surround operations [Itti et al., 1998]. In this thesis I implement this operation by making use of a 2D-array of silicon neurons with excitatory and inhibitory synapses. The synaptic weights are realized with the help of a probabilistic mapping device. The selective attention model scans sequentially the regions of high saliency. I implement this operation by using a neuromorphic chip implementing a 2D-Winner–Take–All (WTA) network with Inhibition of Return (IOR) functionality. In order to control the operational parameters of the neuromorphic chips used in this thesis as well as for the communication of the individual chips with a workstation, I developed a custom hardware/software infrastructure. Furthermore, I present results of experiments conducted with the visual selective attention system to show its functionality.

By implementing the bio-inspired method of selective attention a mobile robot can better assign its computational resources to certain regions in the robot's visual input space. It is the first time that such an implementation of a selective attention system based on a neuromorphic

multi-chip system is presented.

Zusammenfassung

Sehen ist eines unserer wichtigsten Sinne. Um zu sehen, wandelt unser Auge das Licht in elektrische Impulse um. Das Gehirn extrahiert daraus relevante Informationen. Um einen mobilen Roboter mit der Fähigkeit des Sehens auszustatten, müssen beide Verarbeitungsschritte implementiert werden. Mittels heutiger Fertigungstechniken ist es möglich, sparsame Bildsensoren mit hoher Auflösung und hoher Aufnahmefrequenz zu bauen. Sie liefern scharfe, detailreiche Bilder der Umgebung des Roboters. Damit wird der erste Aspekt des Sehens ermöglicht. Der zweite ist, die relevanten Informationen aus den Bilddaten zu extrahieren. Informationen sind für den Roboter relevant, wenn sie ihm ermöglichen mit seiner Umgebung zu interagieren, z.B. seine Position zu bestimmen oder Hindernissen auszuweichen. Um dies zu gewährleisten, ist es nötig, dass die Bilddaten in Echtzeit verarbeitet werden.

Problematisch dabei ist, mit begrenzten Rechenkapazitäten die für den Roboter relevanten Informationen aus der riesigen Menge an visuellen Daten, die der Sensor zur Verfügung stellt, in Echtzeit zu extrahieren.

In der Biologie lässt sich ein analoges Phänomen beobachten: Unsere Augen liefern ein Vielfaches der Daten, die vom menschlichen Gehirn überhaupt verarbeitet werden können. Dennoch können wir mit unserer Umgebung in Echtzeit interagieren. Den Mechanismus, der die relevanten Informationen aus den Daten, die unsere Augen liefern, extrahiert, wird *selektive Aufmerksamkeit* genannt [Treisman and Gelade, 1980]: Nur ein Teil der Bilddaten wird im Detail verarbeitet, der Rest wird verworfen. Ein vorverarbeitendes System erkennt Bereiche im Sehfeld, die salient sind. Die Bilddaten dieser Bereiche werden von unserem Gehirn in sequentieller Weise weiterverarbeitet. Eine Möglichkeit das Datenproblem zu lösen, ist, diese Strategie aus der Biologie auf die Robotik zu übertragen. In dieser Arbeit beschreibe ich ein neuromorphisches Multi-Chip-System, dass von einem selektiven Aufmerksamkeitsmodell abgeleitet ist [Koch and Ullman, 1985]. Das neuromorphische Multi-Chip-System nutzt dem Gehirn nachempfundene Bausteine: Emulationen von Neuronen und Synapsen. Dadurch kann das System Berechnungen sehr effizient ausführen. Um die relevanten Bereiche im Eingangsbild zu erkennen, nutzt das Modell eine Salienzkarte. Diese Karte ordnet jedem Pixel des Eingangsbildes einen Wert für dessen Salienz zu. Im Modell wird die Salienz mittels Zentrum-Umfeld-Funktion bestimmt [Itti et al., 1998]. In dieser Arbeit implementiere ich diese Funktion mit Hilfe eines neuromorphen Chips bei dem künstliche Neuronen und erregende und hemmende Synapsen in zwei Dimensionen angeordnet sind. Die Gewichte der Synapsen werden mittels eines wahrscheinlichkeitsgesteuerten Mappers realisiert. Das Aufmerksamkeitsmodell scannt sequentiell alle Bereiche mit hoher Salienz. Ich implementiere diese Funktion mit Hilfe eines zweidimensionalen Winner-Takes-All Netzwerkes mit Inhibition of Return Funktionalität. Um zum Einen die Betriebsparameter der neuromorphen Chips einzustellen und zum Anderen die Kommunikation zwischen den Chips und einem Computer zu ermöglichen, habe ich eine Hardware/Software Infrastruktur entwickelt. Ausserdem beschreibe ich die Resultate

von Experimenten, die mit dem selektiven Aufmerksamkeitssystem durchgeführt wurden, um seine Funktionsweise zu zeigen.

Durch die Anwendung der selektiven Aufmerksamkeitsmethode, die von der Biologie abgeleitet ist, können mobile Roboter ihre Rechenkapazitäten besser den relevanten Bereichen im Sichtbereich zuordnen. Das hier dargestellte System ist das erste selektive Aufmerksamkeitssystem, dass auf einem neuromorphen Multi-Chip-System aufbaut.

Contents

List of Figures

List of Tables

1 Introduction

Even if our brain's whole computational capacity would be assigned to the visual system it could not process all the information provided by our retinas quickly enough to allow us to interact with our environment in real time [Tsotsos, 2005, Rensink et al., 1997]. Nevertheless, humans are able to react to visual stimuli within a few hundred milliseconds [Posner et al., 1980]. What is the brain's method to circumvent its limited resources but still allow us to interact in real time with our environment? The method is called *selective attention*: A preprocessing mechanism selects parts of the visual scene that are salient. Only salient regions are then processed by higher level brain areas. Usually more than one region in the visual scene is considered as salient. Hence the preprocessing mechanism is able to select more than one region and guide the higher level regions in a sequential manner through its selection. Salient regions are parts of the visual scene that are important to the observer. The importance is determined by the appearance and relation of visual features such as color, motion, orientation, size and others [Wolfe and Horowitz, 2004]. For example on the road red stop lights in front of us in a rainy and therefore mainly gray day will attract our attention. In these cases the saliency is driven from the input to the visual system. This is referred to the term "bottom-up" in the literature. The importance of a region does not only arise from the visual scene itself but is strongly influenced by the context of a task the observer performs. If we are looking for a friend at the crowded train station and know that she wears a red hat we bias our attention system such that we will recognize red objects more easily. This possibility to influence our attention system is referred to the term "top-down" [Connor et al., 2004].

Similar problems arise when constructing a vision systems for a robot to interact with its surrounding environment in real time: With today's technology it is easy to equip even small mobile devices with visual sensors able to provide huge amounts of data. But the processing of this data exceeds the available computational resources by orders of magnitudes. Especially in mobile applications the main constraints are due to the need of low power consumption and/or space restrictions. The problem of such devices is to identify regions within their visual space to which it is beneficial for them to allocate the limited processing resources to perform computationally expensive algorithms like character or face detection. The application of the strategy found in biology is a possible solution to that problem. The goal of this thesis is to implement a low power system using the described method inspired by biology to guide a technical system to locations for further investigations.

1.1 Models of selective attention

Several models of visual selective attention have been proposed during the last years. In this context one has to consider different issues and experimental paradigms: "space vs. object-

based attention, filtering tasks and visual search" [Heinke and Humphreys, 2005]. In this case the goal is to find locations in the visual space which are salient. Therefore computational models performing a space based visual search algorithm are of interest.

Many models describe attention as a spotlight that "shines" light onto a spatial defined part of the visual scene's computational representation leaving the non-attended parts in "shade". Strong support for such models comes from experiments that show that target detection times decrease when a valid cue is presented in advance [Eriksen and Hoffman, 1972, Posner et al., 1980].

One of the most influential models following these lines is the "feature integration theory" developed by Treisman and Gelade [1980]. The authors assume that visual perception is devided into two functionally independent and sequential stages. The first processing stages processes features and is preattentive. Features are values on different dimensions like color, orientation, spatial frequency etc. All features are processed in parallel, automatic and independently across the whole visual space. In a second processing stage different features at an attended location are combined and integrated to form objects.

Based on this theory two important streams of models were developed. Wolfe et al. propose a model for the visual search problem whereas several authors developed a series of models based on a proposal by Koch and Ullman.

Wolfe et al. [1989] cite several other authors and carry out a series of visual search experiments by themselves where the reaction time of subjects does not increase as expected with the number of distractors. Based on these observations they propose a model where the parallel and the serial stages are not completely separate: in cases where conjunctions of features define a target the parallel processing stage cannot distinguish distractors from targets. In the proposed model the result of the parallel stage's calculation is used to divide the objects into possible candidates for targets and distractors. Then the serial stage picks the target out of the candidates. This is much more efficient than picking the target out of all available objects. For this to work the parallel stage needs to know about the properties of the target object. This model was extended and optimized several times to be able to predict a even broader variety of experiments by the authors: Wolfe [1994], Wolfe and Gancarz [1997], Wolfe [2001, 2006].

Based on Treisman's "feature integration theory" Koch and Ullman [1985] proposed a visual attention model: One of the authors' goal was to develop a model that can be implemented in a biologically plausible way. As Fig. 1.1 shows the model describes two stages: According to Treisman's theory the first stage performs operation on the whole visual space in parallel: For visual features like e.g. color, orientation and light intensity the first stage analyzes the whole visual scene. The output of this computation is a map per feature. These feature maps are combined into a saliency map. For each location in the visual scene this global map encodes the conspicuity. The model's second stage selects the location with highest conspicuity values from the saliency map. Koch and Ullman propose a network that performs a maximum operation on the map using "biologically motivated assumptions" [Koch and Ullman, 1985]. The result of the Winner-Take-All network is then the location within the visual scene with highest conspicuity. This location is transmitted to the central representation. Since only one location at a time can be selected and transmitted to the central representation this stage works sequentially.

Itti et al. [1998] describe a detailed implementation of Koch and Ullman [1985]'s model

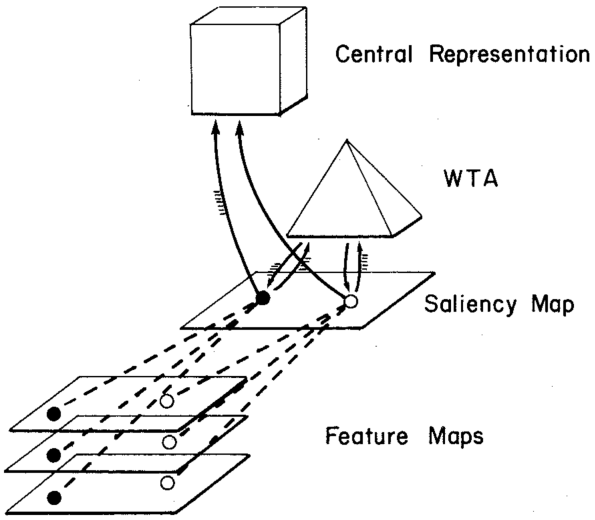


Figure 1.1: Schematic drawing of the visual attention model by Koch and Ullman [1985]. The calculation of the feature maps as well as the saliency map is executed in parallel. The WTA networks picks the region with highest values of the saliency map and forwards its location to the central representation. (from Koch and Ullman [1985])

(see Fig. 3.1). Their attention model focus less on the biological plausibility but rather on the implementation details for simulations executed on computers. The model describes three processing stages: First, color, intensity, and orientation filters are applied to an image at different scales. In a second step, center-surround operations identify spatial discontinuities for each modality. The output of these operations is then combined to create a saliency map. Finally, a WTA network selects from the saliency map the region with maximum values. One method to not only select the region with highest saliency values but also other regions with high saliency values is IOR: After the WTA network chose a region its saliency values are set to zero. Then the WTA network will choose the region with the second highest saliency values. After its selection, again, this region is inhibited in the saliency map to be able to choose the third most salient region. By this process a so-called scan path is created: The WTA chooses sequentially the regions with highest saliency values in a decreasing order. This model is used extensively and different details were subject for further investigation during the last recent years (e.g. Itti and Koch [2001b]). The model is often used as a basis for different extensions (e.g. Frintrop [2006]).

1.2 Selective attention models are implemented using different technologies

There are usually several different means to implement such models. Especially at the beginning of the development – to build a first proof of concept – a software implementation on workstations is considered. This solution stands out due to its flexibility: almost any computational problem can be solved by either using software packages or by writing own programs.

Especially programming languages like Python [van Rossum, 1995] or MATLAB are feasible because they combine advanced computationally powerful built-in functions with an easy to learn syntax. Furthermore, today's computers provide enough power to run complex models as well as the possibility to trace back problems in their implementation. Example software implementations of Itti et al. [1998]'s model are iNVT[1] [Itti, 2004] using C++ or the Saliency Toolbox[2] [Walther, 2006] using MATLAB were developed during the last years. VOCUS [Frintrop, 2006] is another example of a software implementation using a different computation for the combination of intensity and color maps.

Later in development when a working proof of concept should be put into its "production" environment workstation are often too bulky and demand too much power especially if talking about mobile environments. Hence other approaches have to be considered. There are several possibilities: from implementations using microcontrollers via approaches based on Field Programmable Gate Arrays (FPGAs) to solutions with fully custom made Very Large Scale Integration (VLSI) devices. All these approaches have advantages and disadvantages. If the time of development is in focus usually solutions based on microcontrollers are quicker implemented than solutions based on FPGAs or a fully customized VLSI design. The customized design offers other advantages: Since only the functionalities that are really needed are included on such a device the design can be optimized for low power consumption. Considering computational power solutions based on a fully customized VLSI design achieves usually greater operation speeds than FPGA or microcontroller based solutions. These comparisons are true for digital implementations. When a solution based on a fully customized VLSI design is considered also analog or mixed analog/digital implementations are possible.

Brajovic and Kanade [1998] developed an analog two dimensional visual sensor implemented in VLSI technology that is able to track targets. The chip chooses from its visual input space the region with highest light intensity and locks on it. When the source of the light moves the chip is able to follow the target and reports its position in chip coordinates constantly. The authors argue that their chip implements a "rudimentary visual attention system"[Brajovic and Kanade, 1998]: It chooses objects based on their high light intensity which is one of several possible saliency feature in many models of visual attention.

Morris and DeWeerth [1997] designed similar devices: In their paper they present a one dimensional 20 pixel attention system implemented using analog VLSI technology. Because Brajovic and Kanade [1998] are more interested in a reliable tracking than a biological inspired attention system the authors did not implement the inhibition of a selected target as suggested by Koch and Ullman [1985]. Morris and DeWeerth [1997] implemented circuits that perform the Inhibition of Return (IOR) operation. In Morris et al. [1998] the authors extend their approach by being able to not only perform selective attention on a pixel but on a object basis. Therefore their chip groups several pixel to a larger pixel that is entitled as an object. Finally, the same group also developed a two dimensional version of their chip that is presented in Wilson et al. [1999].

The two mentioned approaches share as common property: they are both based on analog VLSI technology. Ouerhani et al. [2002] claim to present the first full implementation of the

[1] http://ilab.usc.edu/toolkit
[2] http://www.saliencytoolbox.net

visual attention model by Itti et al. [1998] in hardware: They present a Single Instruction Multiple Data (SIMD) analog/digital chip able to calculate feature maps at different scales, executing the center-surround operation and fusing the maps to a saliency map. Instead of implementing an IOR operation their implementation selects the most salient regions at once and transfer them via a Direct Memory Access (DMA) interface to a workstation. Further details to their implementation is described in Ouerhani and Hügli [2003].

An emerging field of technology that is in particular suited for systems with limitations in space and/or power is Neuromorphic Engineering [Mead, 1989]. As the name suggests Neuromorphic Engineering tries to emulate neurons with the help of electronic devices. Neuromorphic VLSI chips make use of transistors operating in the subthreshold regime to build low-power neuronal processing systems that emulate physical properties of biological neurons. Neuromorphic chips contain therefore electronic analogs of synapses and neurons that use spikes for both computation and communication. The latter is implemented by routing these spikes by a digital, asynchronous event based bus among chips and workstations.

This technology was also used to implement visual attention systems: Based on his work presented in Indiveri [1999] and Indiveri [2000b] the author presents a stand alone two dimensional neuromorphic attention chip [Indiveri, 2000a]. Instead of the work presented above this chip does not incorporate a sensor itself – it needs sensory input from external sources interfaced with the Address Event Representation (AER) bus. This allows to use the attention chip with other sensory modalities than only visual. The chip is comprised of 8×8 cells interconnected with a WTA network. This network chooses the cell with the highest input amongst the others. The WTA current is fed to the neuronal circuit connected to the winning cell. The generated events are sent via the AER bus off chip. Also the IOR feature is implemented by making use of the cell's output neuron. Based on this work Bartolozzi [2007] developed a chip similar to the described one but with a larger number of cells (32×32). This chip is also used in this thesis.

1.3 The problem addressed: a neuromorphic implementation of a selective attention system

As described in the previous section neuromorphic engineering was already used to implement parts of an visual selective attention model [Indiveri, 1999, 2000a,b, Bartolozzi, 2007]. All these systems are based on the theoretical attention model proposed by Itti et al. [1998]. They implement the model's last stage and hence expect a saliency map as input.

In this thesis I extend previous approaches by implementing also the second stage – the center-surround operations – with the help of neuromorphic chips. Therefore, I build on the work of Bartolozzi [2007] to extend the infrastructure necessary to operate neuromorphic chips and incorporate a second neuromorphic chip within the system. Combining the creation of the saliency map with the existing mechanisms for choosing salient regions from the saliency map creates a selective attention system fully implemented with neuromorphic hardware.

This implementation is important for several reasons: It shows that it is possible to im-

plement a model with several stages with neuromorphic hardware. Furthermore the implementation shows that it is possible to use neuromorphic hardware for a useful task, namely to identify and localize salient regions in a visual input space in real time with low-power devices.

1.4 Thesis outline

This thesis is structured in three parts. In Chap. 2 I describe the toolkit used to implement the proposed neuromorphic selective attention system. It consists of a device that is able to set parameters and enables the communication for the neuromorphic chips. Furthermore the communication infrastructure that connects multiple neuromorphic device and workstations is addressed. Finally, I describe briefly the used neuromorphic devices themselves. In Chap. 3 I describe the model on which the system is based on in detail. Then I reason about the limitations of the presented attention system given that the sensor used is not able to detect color and static images. Then I evaluate theories of center-surround operations that researchers of cat's retina proposed. Even though their insights were collected in the eye I use their results to implement the necessary operations for my selective attention system. In Chap. 4 I describe experiments conducted. My approach is to test first the building blocks of the neuromorphic system and then show the results obtained by combining the different pieces. The thesis ends with a conclusion to summarize the findings and point out the most important achievements.

2 Neuromorphic VLSI infrastructure necessary for the selective attention system's implementation

A broad variety of neuromorphic hardware, such as neuromorphic chips, neuromorphic sensors, and their controlling infrastructure, was developed at the Institute of Neuroinformatics (INI) recently. The following chapter introduces the hardware that is used throughout this thesis, and presents my contributions to the infrastructure developed to use neuromorphic chips and systems. I describe how a single neuromorphic chip can be put into operation. Because I use more than one neuromorphic device I will introduce the communication infrastructure that connects chips and workstations, also starting from a single chip, to multi-chip setups.

2.1 How are neuromorphic devices controlled?

Neuromorphic Very Large Scale Integration (VLSI) chips make use of transistors operating in the subthreshold domain to build low-power neuronal processing systems that emulate physical properties of biological neurons. Neuromorphic chips contain therefore electronic analogs of synapses and neurons that use spikes for both computation and communication [Mead, 1989].

2.1.1 Controlling a single neuromorphic chip: the AMDA board

The core idea of a neuromorphic chip is to emulate biological neurons and synapses with mixed analog/digital circuits in silicon [Mead, 1989]. The neuromorphic chips' parameters, such as the synapses' time constants and weights or the neurons' leaks or refractory periods, can be adjusted. This is necessary because at design time, not all parameters are known and it allows to use the chips for several applications. Changing a parameter of an electronic circuit means changing a voltage or a current on the device. Because it is easier to control voltages, current parameters are usually translated into voltage biases by using a current source on the device controlled by a voltage. The question of how few tens of parameters on a small chip can be controlled can be answered in several ways. One possibility is to put the chip on a Printed Circuit Board (PCB) equipped with controllable voltage sources such as potentiometers or Digital–to–Analog Converters (DACs) and connect their outputs with the input pads of the chip. Another possibility is to equip the chip with a bias generator, a circuitry that generates the necessary analog voltages on-chip [Delbruck and Van Schaik, 2005]. The information

of which voltage values to generate are transmitted to the bias generator by a simple digital interface, the Serial Peripheral Interconnect (SPI) bus.

Besides the parameters of a neuromorphic chip there is the problem of sending and receiving input and output signals to and from the chip. In the majority of animals the neurons communicate amongst each other with short electronic pulses called spikes. This communication method is also emulated by neuromorphic chips: Both, the input and the output signals to the neuromorphic chips are implemented digitally to mimic the spikes in biology. In contrast to an analog signal where a value is encoded e.g. by a voltage that lies between ground and the supply voltage, a digital signal is encoded in a binary fashion by a number of lines where the voltage for a single line can only be either ground or the supply voltage (high). If the line's voltage lies in between it is either considered invalid or interpreted either as ground or high. The means of connecting the digital lines of two different communication partners together is called a bus. One bus used to connect neuromorphic chips is called Address Event Representation (AER) bus.

Because both tasks, sending and receiving of in- and output signals as well as controlling the chip's parameters, have to be solved for almost every neuromorphic chip a standardized mean of providing these functionalities helps avoiding repeated development. Hence a PCB providing the necessary environment, meaning supply voltage, voltages controlling the neuromorphic chip's parameters, and the necessary communication interfaces for in- and outputs, was developed at INI: the AER Motherboard with D/A converters (AMDA). The idea is to have an advanced basis board with all necessary equipment with an easy to implement interface to different neuromorphic chips. Then after designing a new neuromorphic chip one can create a typically simple PCB – a so called daughter board – for this particular neuromorphic chip that can be put on the basis board. The interface between the AMDA and its daughter boards consists of three 64-pin sockets. Their purpose is to carry the daughter board and ensures the electrical connections between both PCBs. Note that typically only one daughter board is put on one AMDA board.

To create the necessary environment for neuromorphic chips the AMDA board is equipped with several devices: For the in- and outputs to and from the chip, the board provides two 20-pin connectors implementing the AER bus controlled by a Complex Programmable Logic Device (CPLD). The different voltages for the different parameters are created by 96 DACs. Analog voltages can be converted into digital values by eight Analog–to–Digital Converters (ADCs). Furthermore it is possible to set four digital pins and read from four digital pins. The first can be used e.g. to transmit voltage values to a built-in bias generator. All these components run at 3.3 V and are powered via an external power supply at 5 V.

How are all these electronic devices on the AMDA board controlled so that they can in turn control the parameters of the neuromorphic chip? The solution is a microcontroller sitting on the AMDA board with a connection to the user's workstation. On the one hand the microcontroller – an Atmel Mega128 – communicates with the electric components providing the environment for the neuromorphic chip via optical couplers. They separate the electric components electrically from the microcontroller to avoid crosstalk and other disturbing electrical influences between the purely digital microcontroller realm and the part of the AMDA board providing the neuromorphic chip's environment. On the other hand, the microcontroller is connected via a Serial-to-USB converter to a workstation from where the users sends com-

mands to adjust voltages and receives information about the state of the neuromorphic chip's environment. The second purpose of the connection between workstation and microcontroller is to provide its necessary power. Whenever the USB connector of an AMDA board is plugged into a Linux workstation a service program (udevd) reads the AMDA board's ID and creates a device file of the form `/dev/amdaboard`*`ID`* where *`ID`* is replaced with the actual board ID. First, this methods makes it easy for programs running on the workstation to identify connected AMDA boards. Second, this device file is the interface for other programs to control the board's functions. If a board is disconnected, the same service program ensures that the generated device file is removed from the workstation's file system.

As part of my PhD project, I developed the firmware that runs on the AMDA microcontroller to manage the environment for the neuromorphic chips (see Appendix A). The principle design objective pursued during developing this software was to keep its complexity as low as possible and to move higher level functions (such as controlling the bias generator) to the controlling workstation. The firmware receives commands from the workstation and parses them. Depending on the command the necessary electric components to perform the requested action are then activated. After performing an action the firmware sends a reply message back to the workstation to confirm the action's execution. In case of an error an error message is sent back. Because each command sent results in a response message only one program can reliably communicate with the AMDA boards via its device file: If more programs read and write messages from and to one device file it is possible that one reads the response of the other and gets wrong values or assumes a malfunction. A possible solution to this problem is presented in the next section.

AMDA – server

The interface to the AMDA board seen from the workstation is a device file. In a common use case, a user would set up a GUI to visualize the bias voltages applied to the neuromorphic chip connected to the AMDA board. At the same time the user might run additional scripts that load predefined bias values to the chip and carry out experiments. Because both happen in parallel several programs would attempt to read and write commands to the same device file. Therefore, chances of interference are high. To manage such use cases, a possible solution is that only one program writes and reads messages to the device file and provides means to other programs to communicate with the AMDA board indirectly via this program. Because all messages have to be processed by this program it is able to handle possible interferences. Such a program is called a server program and the programs that communicate with this server are called clients (compare Fig. 2.1). This solution is a common paradigm in the world of information technology. In collaboration with other members of the team that use this infrastructure I developed such a client-server system.

The server is implemented using the programming language Python [van Rossum, 1995]. Clients communicate with the server via a TCP network connection. Therefore the client programs do not have to run on the workstation connected to the AMDA board but can be located anywhere in the network. This allows that users can work with the hardware controlled by this AMDA server remotely. Another advantage of using a standard network interface is that clients can be programmed in any programming language that supports handling network con-

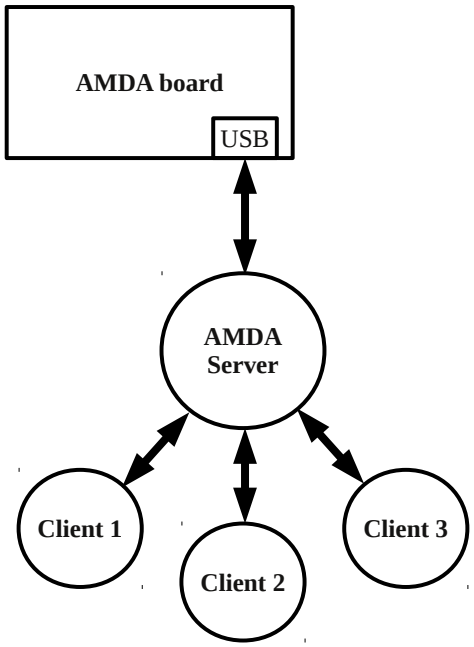


Figure 2.1: To allow several programs to communicate with the AMDA board, a client-server architecture is used.

nections. Besides providing a single communication channel to the AMDA board(s) attached to it, e.g. to transmit new bias voltages values, the server also makes sure that values set by one client are distributed also to the other clients registered for the same board. With this method it is ensured that all clients operate always with up-to-date values.

To make it more convenient for the users the server starts whenever the first AMDA board is connected to a workstation. The same service program that creates the device files for the AMDA boards checks whether an AMDA server runs on the machine. If not the service program starts an instance of the server. If a server is already running it sends a message to the server requesting it to update its internal database with the newly connected AMDA board. When a AMDA board is disconnected from the server the service program also informs the server to remove the unlinked board from the database. This automatic update mechanism guaranties that the server is always "aware" of all connected AMDA boards that can be accessed by client programs.

To use neuromorphic chips they have to be put into a well defined environment: Few tens of parameters have to be set up by appropriate bias voltages. Also the communication to the chip's synapses and the spikes from the neurons has to be ensured. The last section described in detail the means of providing the bias voltages with the help of a standardized PCB, the AMDA board. I highlighted my contributions to this infrastructure in form of the board's firmware and the development of a server-client system allowing several programs to control the environment's bias values at the same time. The next section is going to focus on the second part of the environment by answering the question of how to handle the in- and outputs to and from the neuromorphic chip.

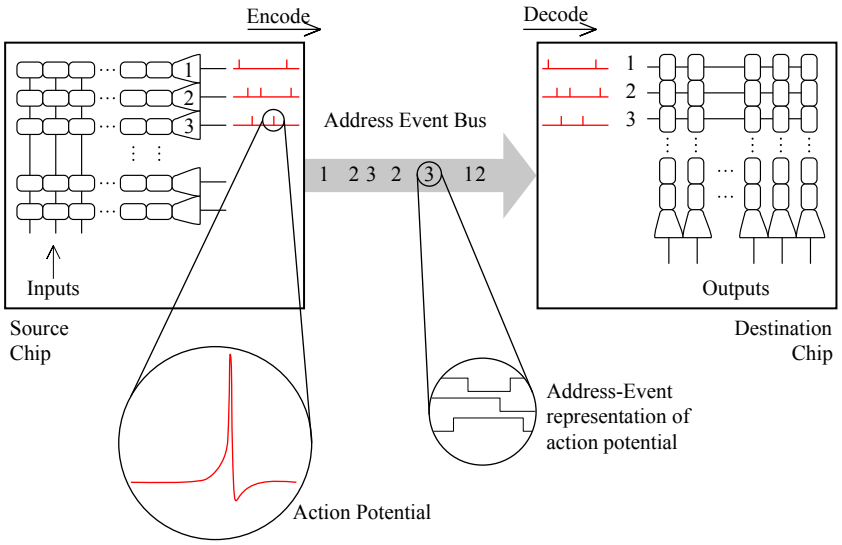


Figure 2.2: Whenever a neuron on the source chip generates an action potential, the neuron's address is transmitted by a digital bus to the destination chip. On the destination chip the addressed synapse generates a current that can be used for further computation. (Adopted from Deiss et al. [1998])

2.1.2 Communication to the neuromorphic device via the AER bus

In biological nervous systems neurons communicate with each other mainly by transmitting short electronic pulses, called action potentials or spikes which are generated at the axon hillock. These spikes are transmitted via the axon to the synapses that connect the neuron with its target neurons. This mechanism is reproduced by the electronic circuits of neuromorphic hardware: whenever the membrane potential of a silicon neuron crosses its threshold voltage it generates an action potential. In biology the spike would then be transmitted by a single wire – the axon – to the target neurons. Because one wants to be able to change the network topology after creating a neuromorphic chip, e.g. to use this chip for different applications with different neuronal networks, a more flexible approach than connecting neurons with wires defined at design time has to be taken (see Fig. 2.2): On the input side each synapse that should be accessible externally is encoded with an unique address. Whenever an address is sent to the chip a circuitry called arbiter resolves the address and generates a pulse at the synapse. Then an acknowledge signal is sent to the sender. Because the arbiter reacts immediately to incoming addresses it is not necessary to encode time separately: whenever a spike is sent it is applied, hence time encodes itself. On the output side, each neuron that should be able to send events to a receiver is encoded by an address. Similar to the input arbiter an output arbiter creates an address value of the neuron that generated the spike and sends the address immediately to a receiver. The receiver response with an acknowledge signal. Just like in biology the activity within neuronal networks working on neuromorphic hardware should be sparse in space and time. Therefore it is an unlikely case that more than one neuron spikes exactly at the same time point. Nevertheless, the output arbiter has to handle this case: it will choose one of the events

to be sent first and will then send the other one. Because the digital circuitry that handles the spike communication works in the range of nanoseconds in contrast to the neurons that work in the order of milliseconds the time shift of an event does not affect the computation of the neurons on the neuromorphic chips considerably.

To connect senders and receivers of addresses a bus is used. Since it is not necessary to encode time explicitly the bus works in an asynchronous fashion: every spike event is immediately transmitted, whenever it is generated. Because only the address of an event emitting neuron is transmitted, the bus is called Address Event Representation (AER) bus. One of the first detailed descriptions can be found in Sivilotti [1991] and Mahowald [1992]. The input and output events to and from the chips uses the parallel AER (pAER) bus. The pAER bus is a straight forward implementation, that uses one wire per bit to transmit, plus one wire for each, request and acknowledge, i.e. the flow control signals.

In a multi-chip setup the communication amongst several neuromorphic chips at a time has to be ensured. Due to more chips producing more events the available bandwidth has to be increased. Therefore a more powerful implementation of AER, serial AER (sAER), was developed [Fasnacht et al., 2008]. Instead of using one wire per bit the sAER uses two pairs of wires to transmit 32 bit addresses serially. One pair is used for the data the other pair transmits flow-control signals. With this approach it is possible to achieve events rates up to 78.125 MHz for 32 bit address events. Because the AMDA boards are not equipped with sAER ports, the boards have to be extended by an AMDA EXtension board (AEX). These extension boards provide both a transmitting and receiving sAER port, two pAER ports and an USB port to communicate with a workstation.

Events are transmitted between AMDA and / or AEX boards in real-time. Since processing on workstations is not real-time capable, events transmitted to a workstation require their timing information to be preserved explicitly. Therefore the AEX has a built-in time counter with a 128 ns resolution. The counter is started as soon as the device file on the workstation is opened. Then the AEX assigns to each event that is transmitted to the workstation a 32 bit timestamp. A Field Programmable Gate Array (FPGA) controls the on-board communication. The AEX boards have to be powered externally with a 5 V power supply or they share the same power supply used by the AMDA board.

AEX – server

A workstation is able to monitor events generated by a neuromorphic chip connected to an AEX through the USB port. At the same time it is possible to stimulate synapses on the setup by sending events generated on the workstation to the AEX. Like the driver of the AMDA board the AEX driver provides a device file on the Linux workstation to provide a method to communicate with the AEX board, i.e. send stimulus events and monitor events. Again, if several programs want to read or write to the AEX, there is a high chance of interference. Similar to the client-server-architecture developed for the AMDA boards, I developed an AEX server to solve this problem. As opposed to the AMDA server, much more data, in form of address events, has to be written and read. Therefore one network port for reading and another one for writing is provide to client programs. This approach also reduces complexity of the clients' development because only the communication to the necessary functionality

(monitoring and/or stimulating), needs to be implemented.

Besides the bias voltages that have to be provided to a neuromorphic chip the communication in form of events to and from the chip has to be established. This is done by different implementations of the AER protocol. To let several client programs on a workstation communicate with one neuromorphic chip at the same time, I developed a client-server system: the AEX server. The address event schema is the basis to define different network topologies. The routing itself is executed by a specialized device that is described in the next section.

2.1.3 Address translating: the AER mapper

A single chip setup consists of an AMDA board carrying the neuromorphic device, and an AEX board that transmits and receives events to and from the workstation. In this type of setup the workstation can generate stimulus events with the necessary addresses and can interpret the received addresses. If more than one chip is used, a device to translate the output addresses from the source chip to the input addresses of the destination chip is required. In Fasnacht and Indiveri [2011] the authors describe a mapper that is able to do this translation. It uses a PCI card equipped with a FPGA to communicate with the AEX boards: it has one sAER port to receive events and another one to send events. The memory of the workstation carrying the PCI card is configured such that it can be used as a look-up table. Whenever the device receives an address event it looks up the destination address(es) in the workstation's memory. Events are then emitted from the mapper to these addresses. The time necessary for this look-up operation is very small compared to the time constants of neurons or synapses: 0.8 µs.

The workstation provides 2 Gib of memory for the mapper's look-up table. For each input address, it is possible to define up to 512 output addresses (see Fig. 2.3). Each output address needs 32 bits to be defined. Therefore the input address space is 20 bits. This is less than the possible 32 bits address space that the AEX boards offer, but it is sufficient for the current setups: the pAER connections from and to the AMDA boards provide only 16 bits! Also the mapper's output address space does not offer the full 32 bit: The first bit is defined as the "stop"-bit. It is used to tell the mapper that this address is the last one to emit. The next seven bits are defined as "probability" bits. If they are all set – corresponding to a value of 127 – the mapper will always emit an event with the given address. If the value is smaller the mapper will only emit an event with a probability of $p = \frac{value}{127}$. Therefore if a received event should not generate any output event the first entry in the corresponding table should be 0x8000 0000. Finally, the remaining 24 bit define the output address.

Based on previous work, I developed a C library to generate the mapping table in the workstation's memory. Based on this library, the tools `setMapping`, `addMapping`, `clearMappingTable`, `printDebugMappingTable`, and `printMappingTable` were developed. Mapping information is saved in regular text files. The mapping tables have an input address and an output address separated by a space per line. In this case, the probability of sending an event to the output address if an event with the corresponding input address was given is set to one. To define a probability, a third column with values between 0 and 127 has to be added to each line.

By putting the mapper into a multi-chip setup a user can manipulate the addresses of events transmitted within the setup. This can be used to the user's advantage to define connections

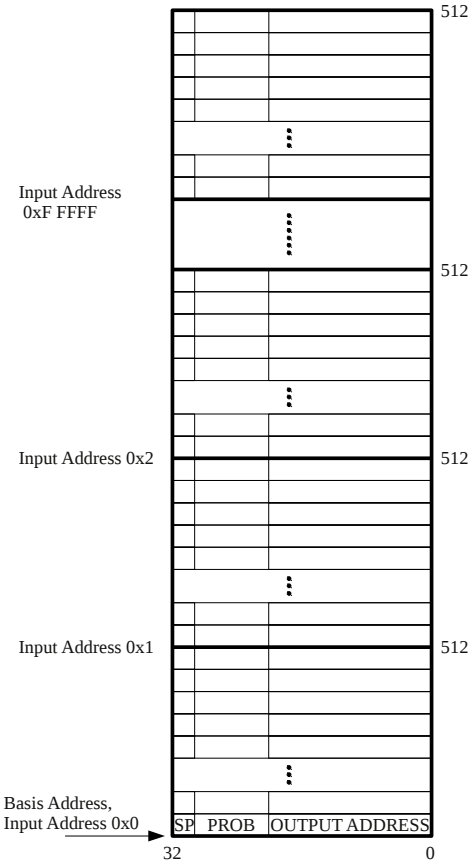


Figure 2.3: The organization of the mapper's look-up table in the workstation's memory. Each entry has 32 bit. The first bit is the "stop"-bit (SP). The next seven bits define the probability (PROB) of emitting an event to the output address. For a given input address, it is possible to define up to 512 output addresses. The maximum input address is 2^{20} (0xF FFFF).

between different neuromorphic chips to specify a network topology.

2.1.4 How do multiple neuromorphic devices communicate amongst each other?

After describing the tools to control the parameters of neuromorphic chips and the devices that enable AER communication, in the following section I describe how these boards can be combined to build a setup consisting of multiple neuromorphic devices. The main question to answer is how several neuromorphic chips with different addressing schemes can be combined in a system to build logical networks of neurons for different applications. I use the term logical network to describe networks of neurons and synapses that a user develops to solve a certain task. In contrast the physical networks are the networks built with the electronic components, the AEX boards and the mapper described in the last sections.

The first question to answer is the physical network topology that the AEX boards and the

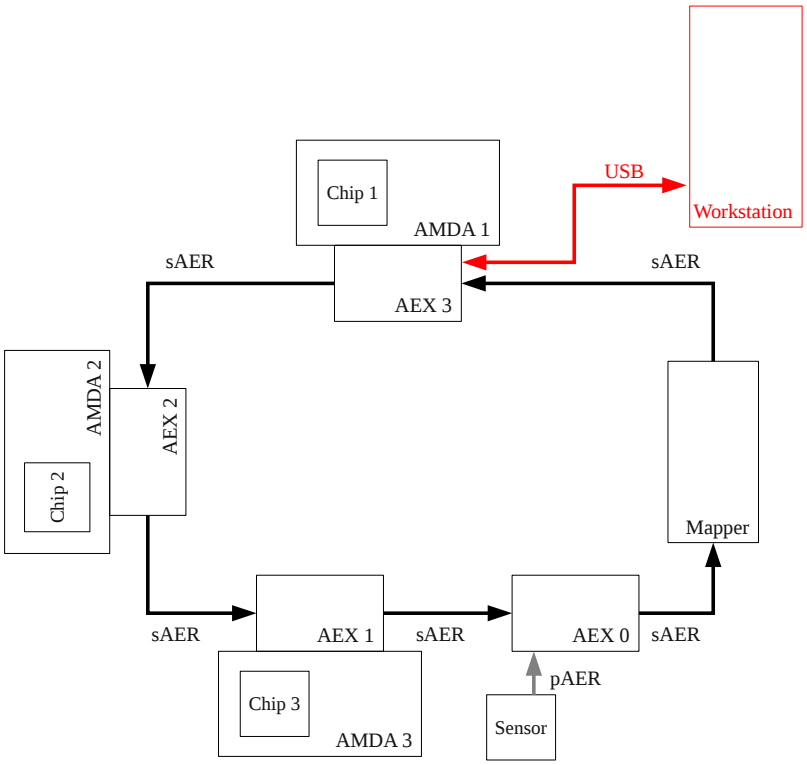


Figure 2.4: General schematic of an example multi-chip setup: Up to four AEX boards and one mapper communicate to each other in a ring structure implemented by sAER links (the arrows show the transmission direction). The AEX 3 board additionally has a USB connection to the controlling workstation. This connection allows two functions: First, one can monitor events in the multi-chip setup. Second, events can be inserted to stimulate synapses on chips in the system. The AMDA boards communicate with their corresponding AEX via pAER (not shown explicitly). Neuromorphic sensors can also be incorporated into the system: they insert their events via a pAER link into the ring as shown on the AEX 0 board. The USB connections from the workstation to the AMDA boards to control the chips' parameters are not shown.

mapper will form. The constraints are the following:

1. Each AEX has one sAER port for receiving and one sAER port for transmitting events. The same is true for the mapper. Because there is only one output port available, there is no choice for the devices amongst several outputs where to send an event: Therefore there is no possibility to route events within the sAER bus.

2. It should be possible for every event in the system, independent of where it was generated, to reach every AEX board so that every event can be sent to every neuromorphic device connected to an AEX board. Additionally, to generate arbitrary logical network topologies with different neuromorphic chips it is necessary that all events in the system pass the mapper since this is the device used to manipulate addresses. Note that the AEX boards are also able to change addresses; in contrast to the mapper where it is easy to change the mapping tables reprogramming an AEX board is time consuming and hence not practicable to use it for establishing logical networks.

With a ring topology (see Fig. 2.4) all constraints can be fulfilled. This means that every device in the ring is connected with one successor. The output of the last one is connected to the first. (Since it is a ring, calling a device first or last is ambiguous anyway.) To monitor and/or insert events into the multi-chip system, one AEX is connected to a workstation via USB.

Each chip uses a specific channel

In an experimental setup with several neuromorphic devices it is necessary to know which device emitted which event and to which device a specific event is targeted to. Since the communication between different neuromorphic devices is achieved by the AEXs, events from/to two different AEXs have to be distinguished. Hence, the addresses generated by the neuromorphic devices are extended by an AEX specific tag. This tag is an ID, i.e. a number, that distinguishes different AEXs. The IDs are shown in Fig. 2.4. Because current neuromorphic devices uses up to 16 bit to address their synapses and neurons, the bits greater than 16 are used for this tagging.

The firmware running on the AEX' FPGA enables the routing between the different communication ports of the AEX. For each of the 6 routes from one port to another the firmware provides a set of filters:

1. Every received event has to pass a filter:

$$a_{min} \leq a \leq a_{max}$$

   where a is the incoming address value, a_{min} a minimal address value and a_{max} a maximum address value.

2. If an event matches both conditions, the following operations are applied:

$$a' = (a \,\&\, m_{and}) \mid m_{or}$$

where a' is the resulting address value that is forwarded, & is the bitwise "AND" operator with the corresponding mask m_{and} and | is the bitwise "OR" operator with the corresponding mask m_{or}.

Given these rules, I developed an addressing schema for an efficient and flexible ring traffic (see Fig. 2.5):

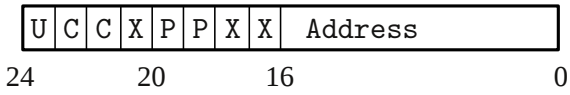


Figure 2.5: Addressing schema: U: USB-flag, C: consumer, P: producer, X: free to user

- USB-flag: The AEX that communicates with the workstation via USB only forwards events to the workstation if this flag is set. Since this flag is at bit position 23, it can only be set by using an appropriate mapping on the mapper (the mapper only accepts events with bit length of 20 or smaller but is able to transmit events with a bit length of 24). With the help of that bit, the user can decide which events she wants to monitor. This is especially helpful when there is a lot of traffic within the ring: since the USB bandwidth is lower than the sAER bandwidth, the user can specify the interesting subset of events to be transmitted to the workstation.

  To achieve this functionality, appropriate minimal and maximal addresses a_{min} and a_{max} and masks m_{and} and m_{or} have to be applied to the route from the incoming sAER-port to the USB-port.

- In bit 21 and 22 up to four target AEXs are encoded. By using appropriate a_{min} and a_{max} for the route from the input sAER-port to the output sAER-port, these two bits can be compared to the AEX' ID. If the event's and the AEX' IDs match, the channel information is removed from the event's address and it is sent to the pAER output port of the AEX. Thereby the event is removed from the traffic within the ring. This reduces the overall traffic within the ring. If the IDs do not match, the event is forwarded to the output sAER-port.

- Whenever an AEX receives an event at its input pAER port, it add its ID into the bits 18 and 19 and inserts the tagged event into the ring. This is done by an appropriate mask m_{or} in the pAER – sAER route.

- A user of the addressing schema is free to use the bits in Fig. 2.5 marked with a 'X' for her own purpose. For example, if it becomes necessary to address more than 4 AEX, then the user could extend the channel bits with bit 20 and bit 17 respectively. In this case bit 16 can be used for other purposes.

By using this addressing schema it is possible to establish almost arbitrary connection between neurons on different neuromorphic devices. Due to the use of the filtering capabilities of the AEX boards it reduces the activity within the ring considerably. Since there are still unused bits in the address space it is possible to extend the schema to be used for up to eight AEX boards.

2.2 The three different neuromorphic devices used in this thesis

The multi-chip setup used in this work comprises of three different neuromorphic devices: The first is the Dynamic Vision Sensor (DVS), the second is a general purpose neuromorphic chip called Integrate & Fire 2–Dimensional WTA (IF2DWTA), and the third is a chip designed to detect maxima in its input: the Selective Attention Chip (SAC).

2.2.1 The DVS: a visual neuromorphic sensor

The DVS is the 128×128 pixel sensor described in Lichtsteiner et al. [2008]. This sensor responds to temporal changes in the logarithm of local image intensity, thus encoding relative temporal changes in contrast, rather than absolute illumination (as in conventional cameras).

Thanks to the logarithmic compression, the DVS is able to detect contrast changes as low as 20 % with a dynamic range spanning over 5 decades. Each pixel in the DVS performs this computation independently (local gain control), allowing the DVS to optimally respond to scenes with non-homogeneous illumination (e.g. outdoors or in environments with uncontrolled illumination). An important feature of the DVS, which makes it radically different from the sensors used in conventional machine vision approaches is the way it transmits output signals: signals are not scanned out on a frame-by-frame basis. Rather, the address of a pixel is transmitted on a shared digital bus, as soon as that pixel senses a difference in contrast. This "event" is written on the bus as it happens, in a completely asynchronous fashion. Each pixel address is written on the AER bus in real time, and potential conflicts (cases in which multiple pixels attempt to access the shared bus at the same time) are managed by an on-chip arbiter. As the DVS only transmits data when pixels sense sufficient contrast changes, redundancy in the data is strongly reduced (e.g. no data is transmitted and no bandwidth is used when there is no change in the visual scene). This produces a sparse image coding and optimizes the use of the communication channel, as well as the post-processing and storage effort. This, combined with the real-time asynchronous output nature of the DVS ensures precise timing information and low latency [Lichtsteiner et al., 2008] yet requires a much lower bandwidth than used by frame-based image sensors of equivalent time resolution [Delbruck, 2008].

The sensor is mounted on a PCB developed by Ángel Jiménez that provides access to the sensor's pAER bus. Therefore it can be easily incorporated into AER based hardware communication infrastructures.

2.2.2 A "general purpose" neuromorphic chip: the IF2DWTA

The IF2DWTA chip was developed by Elisabetta Chicca and Giacomo Indiveri. It contains a sheet of 32×64 neurons. Each neuron is equipped with ten synapses, of which three can be accessed via AER. Two of the accessible synapses are excitatory, whereas the third is implemented as an inhibitory synapse. One synapse is connected recurrently to the neuron's output; the remaining six receive input from the nearest neighbors in both, x- and y-direction and the second nearest neighbors in y- direction (see Fig. 2.6). All synapses of one group,

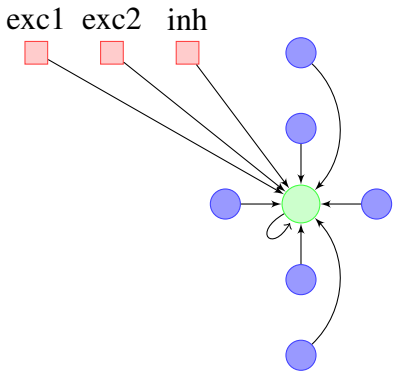


Figure 2.6: The built-in connections of the IF2DWTA: every neuron is recurrently connected to itself. Additionally it receives input from its first nearest neighbors in both, x- and y- direction as well as the second nearest neighbors in y-direction. Finally, it can receive input from synapses that can be targeted from the AER bus: two are excitatory, one inhibitory. Red squares represent AER accessible synapses, blue circles neighboring neurons, and the green circle is the considered neuron.

e.g. the inhibitory synapses, share one weight that can be set via the AMDA board. Therefore the chip has ten different bias values for the ten groups of synapses.

The IF2DWTA was developed to provide a two dimensional Winner–Take–All (WTA) chip that is implemented by using a sheet of spiking neurons. By decoupling of some of the built-in connections, it is also possible to use the chip for several one dimensional WTAs. If all built-in connection strenths are set to zero, all neurons work independently. With the help of AER, the chip can then be used for any neuronal network that needs not more than two excitatory synapses and one inhibitory synapse per neuron.

Due to a bug in the acknowledge lines of the inhibitory synapses only half of them can be used: Only the synapses in the columns 0, 1, 4, 5, 8, 9, ..., 28, 29 work reliably.

The IF2DWTA was fabricated using a standard AMS 0.35 µm CMOS process, and covers an area of about 15 mm^2.

2.2.3 A "selective attention chip": the SAC

The SAC was developed to implement part of Itti et al. [1998]'s model of attention. Chiara Bartolozzi described it in detail in her PhD thesis [Bartolozzi, 2007].

The chip is comprised of an array of 32×32 cells with AER digital circuits as well as analog neuromorphic circuits that implement silicon synapses, neurons, and additional signal processing stages. Figure 2.7(b) shows the block diagram of an SAC cell: each cell in the array receives input sequences of spikes; an input excitatory synapse integrates the spikes into an excitatory current I_{exc} which is then fed into a hysteretic WTA circuit [Indiveri, 2001]. The hysteretic WTA network compares the input currents of all cells and activates only the cell receiving the largest input current, while suppressing the output of all other cells. The winning cell will then produce a constant output current I_{wta}, which is independent of the

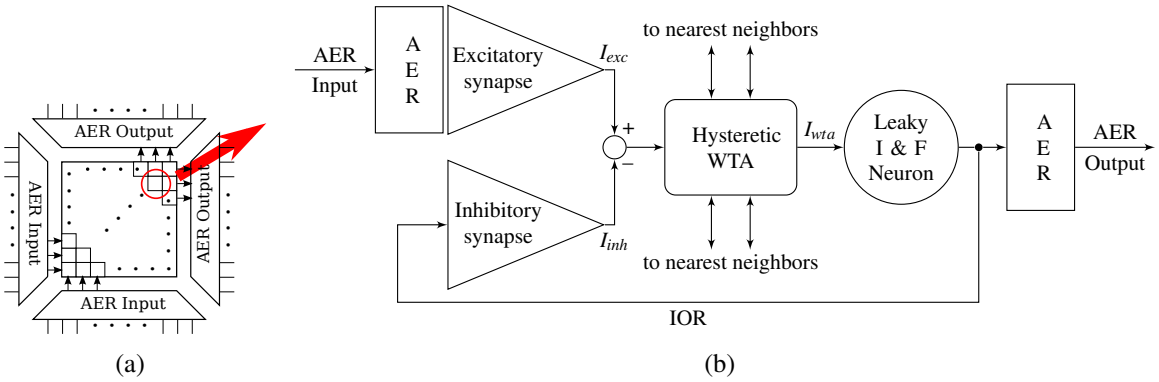


Figure 2.7: SAC diagram. (a) The SAC consists an array of 32×32 cells providing its computational resources and communicates with other hardware via AER receiver-transmitter circuits. (b) Block diagram of one SAC cell. Each cell receives AER spikes from the input bus and competes for saliency by means of a hysteretic winner-take-all network connected to its neighbors via lateral connections. The winning cell sends its address to the output AER bus and self-inhibits via a local inhibitory synapse. All blocks are implemented with hybrid analog/digital circuits described in Bartolozzi and Indiveri [2009].

input, and source it to the cell's leaky Integrate & Fire (I&F) neuron. This circuit, fully characterized in Indiveri et al. [2006], produces voltage pulses (spikes) at a rate which is proportional to it's input current. Each time a spike is emitted from a neuron, it's address is sent off-chip via AER. In parallel, the spikes of the I&F neuron are sent to the cell's inhibitory synapse which generates a current I_{inh}. This implements a negative feedback loop in which the current integrated from the output spikes I_{inh} is subtracted from the excitatory input current I_{exc}. The net input current to the winning cell therefore decreases until another cell wins the competition. This self-inhibition implements a known mechanism in selective attention models named Inhibition of Return (IOR). It allows the network to shift from the currently attended stimulus to a different one, selecting sequentially the most active regions of the input space in order of decreasing activity, reproducing the attentional scan path [Itti and Koch, 2001a].

The SAC was fabricated using a standard AMS 0.35 µm CMOS process, and covers an area of about 10 mm^2.

2.3 Conclusion & Discussion

Throughout this chapter I described different hardware and infrastructure that is used for this thesis. I started with the environment for a neuromorphic chip: To let it work few tens of parameters in the form of voltages have to be set. Furthermore input events have to be transmitted to the device and the resulting output has to be handled. For a single neuromorphic chip this can be done by an AMDA board. For this PCB I developed the firmware and a server-client communication infrastructure. If more than one chip should be used the AMDA board has to be extended by an AEX since this board provides a more powerful bus. Together with

a mapper, an address translating device, one can build a multi-chip setup. By following the addressing schema I developed the setup can deal with devices that create events at a high frequency. An example for such a device is the Dynamic Vision Sensor (DVS), a neuromorphic vision sensor that I use in my project. I close this chapter with the descriptions of the other two neuromorphic chips used, the IF2DWTA and the SAC.

Common digital systems use a hardware clock signal to synchronize their computational units. This clock signal is generated centrally and distributed over the whole device. The clock signal itself is in general constantly and regularly alternating between low and high. Each computational unit has to execute (parts) of a calculation between each clock cycle. At the end of a cycle the result has to be presented in a stable fashion to the next following arithmetic unit. This rigid schema has the advantage of relative easy design and deterministic behavior. This advantage comes at great costs: the generation and maintenance of the clock signal is the source of high power dissipation [Gronowski et al., 1998]. The hardware presented in this chapter follows a different design paradigm that can also be observed in the brain: all computational units be it emulated neurons or synapses, WTA-cells, or the AER-logic are not synchronized by a clock signal. They carry out their operations independently and in parallel: The synapses create currents whenever they receive an input spike, the neurons integrate these currents at all time. Whenever the neuron voltage crosses the threshold voltage they generate a spike that is immediately transmitted with the help of the AER-logic. The lack of a clock in the presented hardware is one reason for the high power efficiency.

Another reason is the different representation of values: Digital systems use a binary value representation. Hence they encode values by series of signals that are either low or high. The low and high signals are usually mapped to the ground and the supply voltage respectively. Hence the transistors on these devices are used as switches that drive the signals to either low or high. In the hardware presented throughout this chapter values of e.g. membrane potentials are not represented by a series of binary signals. Neurons and synapses are emulated by circuits. So the neuron's membrane potential is the voltage on the emulated neuron's capacity. To use transistors in these circuits they operate in the sub-threshold regime. In this regime the currents through the transistors are very small. Therefore the power dissipated by the whole devices is very small as well. The disadvantage of using transistors in sub-threshold regime is that the devices are very prone to fabrication mismatch.

The setup presented consists of several PCBs and two workstations. Therefore the system dissipates quite some power even though the chips themselves are very power efficient. The extra equipment is needed mainly for one reason: flexibility. It gives the researcher the possibility to exchange chips easily, experiment with different sets of biases and create different connections in form of mappings between the different devices. If the final system should be put into production all the chips can be integrated onto one PCB or even on one VLSI device. An example system of a highly integrated neuromorphic device is presented in Silver et al. [2007].

The lack of a clock and the use of transistors in sub-threshold regime, are a great advantage over classic digital devices. Nevertheless, these advantages come with downsides in the form of non-trivial design and fabrication mismatch. Systems that are built based on these devices have to find ways to deal with these challenges. The system presented shows that it is still possible to build systems that can carry out non-trivial tasks even though facing the problems

described.

3 From theoretical models of attention to neuromorphic hardware implementations

This chapter describes the model that I used to implement the neuromorphic attention system. In this thesis I implement only parts of model. Despite some restrictions I argue that the system still provides the main properties of an attention system. Furthermore I describe how the center-surround operation, necessary to detect spatial discontinuities, is implemented using neurons and synapses emulated on a neuromorphic chip.

3.1 A saliency-map based visual attention model

Sec. 1.1 gives an overview of several models trying to explain visual selective attention. As the hardware used in this work is the same that was described by Bartolozzi [2007] I am using the same model: "A model of saliency-based visual attention for rapid scene analysis" by Itti et al. [1998]. This model is described in Sec. 1.1 as well.

The model's general structure is shown in Fig. 3.1. Feature maps are created from a static input image. Feature maps contain information about the saliency for a specific feature at different scales. In their model, the authors investigate the features color, intensity and orientation of the input image. The maps are created by applying a center-surround algorithm on the feature channels of the input image. Hence the model is able to detect "local spatial discontinuities" [Itti et al., 1998] that stand out from their surrounding and are therefore salient.

For each feature the maps are combined across-scales. The resulting maps are called conspicuity maps. They contain the information about salient regions within the input image across scales but with respect to a specific feature. Conspicuity maps of different features, like color or orientation, cannot be compared a priori, because they represent different modalities with different dynamic ranges and extraction mechanisms [Itti et al., 1998]. Hence the maps have to be normalized first. In their model, the authors use a normalization method where maps with a small number of peaks in contrast to maps with many high values are promoted. Furthermore, the normalization procedure ensures that the resulting values are in a fixed range. This normalization allows to combine the conspicuity maps to form a global saliency map. The combination of different maps of different features to a global map is still subject of further research [Itti and Koch, 2001b].

The saliency map assigns to each location in the input image a scalar value of the location's saliency. To guide the focus of attention through the input image one wants to select the image's n most salient regions. Therefore the model uses a Winner–Take–All (WTA) mechanism in combination with Inhibition of Return (IOR). The WTA network chooses the location with the highest saliency value. While this information is made available as the system's output

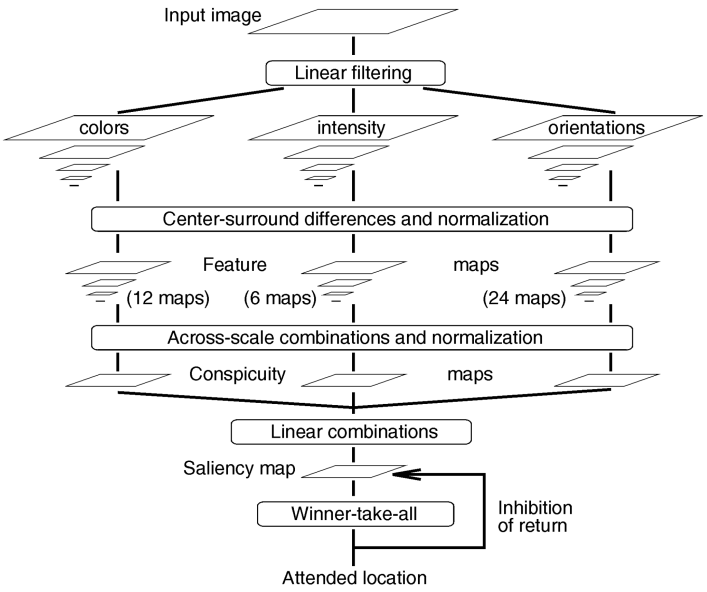


Figure 3.1: General architecture of the saliency-based visual attention model (from Itti et al. [1998])

the region in the saliency map is also inhibited, i.e. the saliency values are turned down for an adjustable period of time. Thereby the WTA will choose the second most salient region which again is made available as output and fed back to the inhibitory system. By choosing appropriate values for the size of the chosen region and the time constants of the inhibitory system, the model is able to create a scan path through the input image.

This model is purely bottom-up driven, since it only relies on the input image. In Itti and Koch [2001b] a possible incorporation of top-down control is proposed.

3.2 Motion is an important selective feature

The visual sensor used throughout this thesis (described in Sec. 2.2.1) is sensitive to "relative changes in intensity, discarding most illuminant information, leaving precisely timed information about object and image motion"[Lichtsteiner et al., 2008]. This means if the sensor does not move it detects only moving, appearing, disappearing or flickering objects in the visual scene. Therefore it cannot be used to capture static scenes and does not detect color.

Given these restrictions is it still justifiable to build a selective attention system by using only the sensory data provided by the DVS? To answer this question I reviewed the literature summarized in this section showing that it is in deed reasonable to build a selective attention system despite the fact that the sensor detects only objects that change their location or appearance.

Let me start with a quote from a paper already pointing out that motion is probably the feature that counts most in biology:

> If asked what aspect of vision means the most to them, a watchmaker may answer 'acuity,' a night flier 'sensitivity,' and an artist 'color.' But to the animals

> which invented the vertebrate eye, and hold the patents on most of the features of the human model, the visual registration of movement was of the greatest importance. [Walls, 1942, p.342]

In a review by Wolfe and Horowitz [2004], the authors create a list of features that contribute to attention. They conclude, that for color, motion, orientation and size there is enough data that there is no doubt that these features guide attention. As a reference they state the study of Dick et al. [1987]. They examined different experiments: First, they showed subjects a display were amongst stationary dots in some trials one dot moved either a short or a long distance. The subject had to detect if there was a motion or not. From their results they concluded that the detection of motion was performed in parallel since reaction times did not increase with increasing number of distractors. The same was true if they let objects appear or disappear on the screen observed by their subjects. Because their experimental setup did not allow them to show real motion but rather a disappearance on one location and an appearance at another location of dots they suspected that the long distance motion could not have been perceived as a motion. Hence they carried out a second set of experiments where subjects had to identify a target – a dot moving to the right – within dots moving to the left. For this set of experiments the authors reported a clear difference between the long range and the short range motion: According to their results, shown in Fig. 3.2, only the short range motion is processed in parallel by the visual system and therefore a real feature for attention. The long range motion, or better stated the combination of a disappearance and an appearance of a dot, is processed serially. Even though the authors found a difference for long range and short range motion other studies like Royden et al. [2001], McLeod et al. [1988], and Nakayama and Silverman [1986] show that in a visual search task motion is processed very efficient and is therefore a feature that guides attention systems.

All the studies showed that motion is a salient feature. Salient features can be detected efficiently, or in other words are processed in parallel, according to the view of Treisman and Gelade [1980]. Yantis and Egeth [1999] distinguish between salient features and features that captures attention in a purely bottom-up manner. By their definition a feature captures attention if in a visual search experiment one can observe attentional effects even if the feature in question is explicitly task irrelevant. With a series of experiments Abrams and Christ [2003] showed that motion onset is an example of such an attention capturing feature. An object that starts moving in the visual scene observed by the DVS generates events. These events can be used by a connected attention system to determine the most salient region.

Itti [2005] compares in his study the recorded saccades of humans while watching video scenes with the output of a saccade predicting computational model [Itti and Koch, 2001b]. His computational model, that is an extension to the one described in Sec. 3.1, uses additionally to color, orientation, and intensity, flicker and motion as features. If he restricts his model to use only one of these features, both flicker and motion perform best. Hence, these two features that are captured by the vision sensor used in this thesis are a good subset of features provided by our eyes to build an attention system.

A fixated DVS detects temporal contrast changes in its visual field. Moving, flickering, appearing and disappearing objects generate these changes. According to the literature reviewed in this section building an attention system relying on the events generated by this sensor is

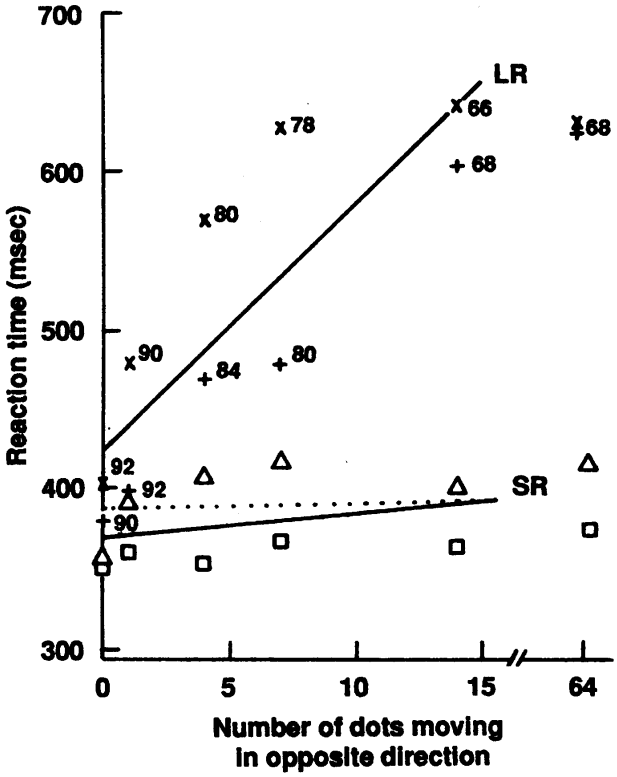
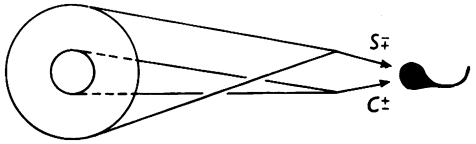


Figure 3.2: Reaction times of two observers for the detection of a target in relation to number of distractors. Target was a moving dot. Distractors were dots moving to the opposite direction. For short moving distances (SR), the detection was processed in parallel, whereas for long moving distances (LR) a serial process took place. (from Dick et al. [1987])

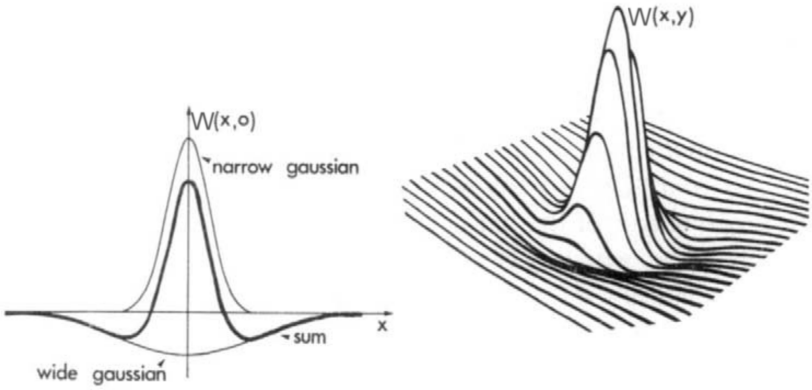
justifiable.

3.3 Calculating the saliency by center-surround operations

Salient regions in the input space are regions that stand out from their surrounding in one or more aspects. The model from Itti et al. [1998] proposes, that these spatial discontinuities can be best identified by center-surround operations easily implementable with neuronal networks. In several visual regions of the brain, such as the retina, the lateral geniculate nucleus, and the primary visual cortex, researchers have found center-surround mechanisms [Leventhal, 1991]. But also in other sensory areas of the brain this concept could be identified [Knudsen and Konishi, 1978, Vucinic et al., 2006]. In the next section, I overview literature leading to models of center-surround operation.



(a) Schematic of center and surround regions



(b) Weight functions

Figure 3.3: Modeling the center-surround receptive field of a retinal ganglion cell.
(a): The ganglion cell receives two antagonistic inputs: one from the center, the other from the surround region (from Enroth-Cugell and Robson [1966]).
(b): (from Rodieck [1965])

3.3.1 Center-surround operation found in the central nervous system

Back in the fifties, Kuffler [1953] investigated the cat's retina and reported that depending on where a small light spot is placed in the receptive field of a ganglion cell, the cell responses with increasing or decreasing activity: If the center of the receptive field is stimulated, the cell's activity rises whereas if the surrounding is stimulated, the cell's activity decreases, then a ganglion cell is called "on"-cell. In contrast, if the neuron's response increases to a stimulus in the surrounding area whereas its activity decreases if its center is stimulated then it is called "off"-cell.

Later, Rodieck and Stone performed a series of experiments again with the cat's retina to investigate the findings of Kuffler not only in a qualitative way, but also in a quantitative way. They found evidence for Kuffler's findings and presented their results in a series of papers: Rodieck and Stone [1965a], Rodieck and Stone [1965b], and Rodieck [1965]. They recorded from 80 ganglion cells and examined the receptive fields of 34 of them. Based on this data, they finally proposed a model for the receptive field of retinal ganglion cells. The ganglion cells receive two antagonistic inputs from two elementary areas: one from the center region and one from the surround region. Both regions are thought to be circular and concentric (see Fig. 3.3(a)).

Rodieck [1965] uses two Gaussians (see Fig. 3.3(b)) to describe the connectivity weights

from the bipolar to the ganglion cell: For "on"-cells, the bipolar cells in the narrow center region project with excitatory weights to the ganglion cells. In contrast, the surround region is much wider. Bipolar cells located in this region inhibit the ganglion cell. For "off"-cells it is just opposite: The bipolar cells located in the center region inhibit the ganglion cell, whereas bipolar cells connected to the surrounding region excite the ganglion cell.

A little later, Enroth-Cugell and Robson [1966] performed similar experiments: Instead of using a small light stimulus, they used sinusoidal gratings presented to anesthetized cats. Just as Rodieck [1965], they described the connectivity weights as a difference of two Gaussians (see [Enroth-Cugell and Robson, 1966, page 535]):

$$W(r) = k_c \exp\left[-\left(\frac{r}{r_c}\right)^2\right] - k_s \exp\left[-\left(\frac{r}{r_s}\right)^2\right] \tag{3.1}$$

Positive weight values $W(r)$ are considered as excitatory weights, negative values are considered to be inhibitory weights. In all parameters, the subscript c stands for the center's parameters, the subscript s for parameters of the surround, respectively. k_c and k_s define the maximum values for the Gaussians, whereas r_c and r_s represent the characteristic radii.

Even though every ganglion cell expresses different parameters, Rodieck reports for a typical unit the estimated parameters $k_c/k_s = 11.25$ and $r_c/r_s = 1/3$, respectively. Enroth-Cugell and Robson [1966] list a table with parameter values for 17 "on"-center cells and 4 "off"-center cells. Their average value for k_c/k_s is 37.4, for r_c/r_s, 0.21. A more recent source for biological plausible parameters for these two ratio is Linsenmeier et al. [1982] and Freed et al. [1992]. Also these values correspond well with the given ones. Nevertheless, all these values were inferred from the retina. Since attention will be computed in much higher brain areas, I looked for estimates for parameters in these regions. In Derrington and Lennie [1984] the authors recorded from neurons in Lateral Geniculate Nucleus (LGN) of the macaque. The parameters reported by them are very similar for the ones reported for ganglion cells ($k_c/k_s = 28.4$ and $r_c/r_s = 0.342$). Even though LGN is higher up in the hierarchy it is not know as a brain area where computation for attention takes places. At the end of the day I am aiming to implement a technical system to perform attention. Therefore there is no technical reason to stick to parameters that were reported in biology. However the question arises in which range should the parameters for k_c, k_s, r_c, and r_s be?

3.3.2 Theoretical consideration of ganglion cell's receptive field's weight parameters

Figure 3.4 shows the four extreme cases for a ganglion cell together with its receptive field. For a ganglion cell, one can derive a general input-output relationship [Dayan and Abbott, 2001]:

$$v = F(I_s)$$

where F is a function relating the synaptic input current I_s to the cell's output frequency v. In the case of a ganglion cell, the input current I_s consists of the current generated by the

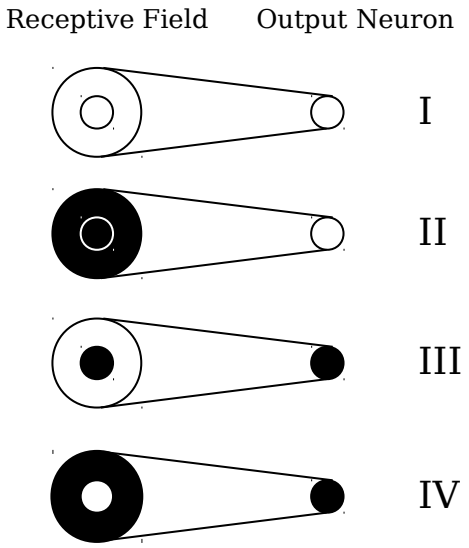


Figure 3.4: Four extreme cases of ganglion cells with their receptive fields. The neuron's receptive field is divided into a center and a surround region. Black indicates high activity, white no activity.

surrounding and the center, both, excitatory e and inhibitory i:

$$I_s = e_s - i_s + e_c - i_c$$

For this simple model I assume a linear relationship from input current to output frequency, so that I can write:

$$v = C \cdot [e_s - i_s + e_c - i_c]_+ \tag{3.2}$$

with C a non-negative constant relating the scaling the input current to the output frequency. $[]_+$ denotes the half-way rectification to ensure non-negative firing rates.

For each of these cases a general input-output relationship can be analyzed:

A spatial discontinuity is something special in the input space in relation to its surrounding. Hence, in the cases (I) and (II), the output neuron should not generate any output, since the input is uniform. In the cases (III) and (IV) there is a difference in the input to the center and the surround regions. Hence, the output neuron should be active. Mathematically, this implies that the input in case (II) result in a zero output.

$$e_c - i_c + e_s - i_s = \frac{v}{C} \stackrel{!}{\leq} 0 \tag{3.3}$$

In the cases (III) and (IV) the output has to be positive, while only the center or the surround region is stimulated (the other terms are zero):

$$e_c - i_c = \frac{v}{C} \stackrel{!}{>} 0 \tag{3.4}$$

$$e_s - i_s = \frac{v}{C} \overset{!}{>} 0 \tag{3.5}$$

Both conditions, 3.4 and 3.5, require a positive value for $\frac{v}{C}$. In condition 3.3 both conditions are summed and should result in a zero value. Because both are positive their sum has to be positive as well. Therefore, there are no possible values for e_c, i_c, e_s, and i_s that could fulfill all conditions at the same time. Hence, there have to be different cells handling the cases (III) and (IV) independently. This is in accordance to biology, where so called "on"- and "off"-cells exist [Rodieck and Stone, 1965a].

The input current to the model I_s can also be seen as integral over all input activities u times the corresponding weight W at all locations:

$$I_s = \int_{-\infty}^{\infty} u(r) \cdot W(r)\,\mathrm{d}r = e_s - i_s + e_c - i_c$$

In case (II) the neuron's whole receptive field is activated equally. Still the neuron is not driven enough to generate an output. From equation 3.3 follows that the input current I_s has to be smaller or equal zero. Since the activity over the whole field is constant and nonzero, the integral over the weights has to be equal or smaller than zero. For the weights I use equation 3.1:

$$\int_{-\infty}^{\infty} W(r)\,\mathrm{d}r = \int_{-\infty}^{\infty} k_c \exp\left[-\left(\frac{r}{r_c}\right)^2\right] - k_s \exp\left[-\left(\frac{r}{r_s}\right)^2\right] \mathrm{d}r \overset{!}{\leq} 0$$

Note that the radii r_c, and r_s are the characteristic parameters for the Gaussians and not the absolute values for the real radii of the center and the surround.

The integral can be solved to:

$$k_c r_c \sqrt{\pi} - k_s r_s \sqrt{\pi} \overset{!}{\leq} 0$$

This give rise to conditions for the parameters of the weight function:

$$k_c r_c \leq k_s r_s \tag{3.6}$$

The relations of the parameters k_c, k_s, r_c, and r_s are used to create mappings for the AER-mapper described in Sec. 2.1.3. The mappings are used to emulate the different weights from the receptive field to the neuron, that calculates the center-surround operation. In Sec. 4.1.2, I show the results of different experiments to investigate the center-surround operation on the neuromorphic hardware.

Comparing these conditions of a center-surround cell reacting to these extreme stimuli shown in Fig. 3.4 with parameters measured from biology, shows that especially for the "on"-cell case, the biological parameters do not meat the derived conditions. The basic consideration that led to these conditions was that the neuron does not response if the whole receptive field is stimulated with the same stimulus (extreme case II). Looking at biological recording, e.g. from Rodieck and Stone [1965a], shows that the ganglion cells are not silent in the mentioned case. Hence the parameter conditions are based on theoretical assumptions that are not

reflected in biology. However, the results from these considerations can be used as hints to choose a working parameter set for an engineered system.

3.4 Conclusion & Discussion

In this chapter I considered three important issues contributing to the attention system. First, I introduced the model of Itti et al. [1998] that is the theoretical basis of my work. Second, I discussed the question if it is justifiable to talk about an implementation of a visual attention system given that the DVS provides only data related to temporal changes in contrast of its visual scene. There is evidence, that motion, motion onset and flicker are basic feature that contribute strongly to the process of determining salient regions in the visual space. Third, I discussed how a central element of the attention model, the center-surround operation, can be implemented in a neuromorphic fashion, that is with synapses and neurons of a neuromorphic chip.

Based on a literature review of studies investigating the center-surround mechanisms in the cat's retina I extracted a formula that can be implemented to carry out center-surround operation on neuromorphic hardware. By examining four extreme cases I showed that it is necessary to implement both, "on"- and "off"-cells, and in which ranges the parameters should be. In the attention model of Itti et al. [1998] center-surround operation are used to detect spatial discontinuities. Hence I investigated how these could be implemented with neurons and synapses as provided by neuromorphic hardware. Therefore I looked for appropriate literature and found the work reviewed in the previous section. The main results were obtained by experiments carried out in the cat's retina. Even though center-surround operations are also found in other sensory modalities [Knudsen and Konishi, 1978, Vucinic et al., 2006] and other brain regions [Leventhal, 1991] I do not claim that in the primate's attention system similar center-surround operations are carried out. I use the knowledge from the retinal literature as input to engineer the center-surround operation proposed by Itti et al. [1998] in the context of my attention system. To conclude: From a biological point of view taking the findings of the cat's retina to "calculate" attention is very questionable. For an engineered system borrowing the eye's technique to locate spatial discontinuities is acceptable.

4 Conducted experiments & their results

In this chapter I describe several experiments I conducted with the system I described in the previous chapters and their results.

First, I focus on experiments that examine the center-surround operations. Then I describe the experiments that were conducted with the Selective Attention Chip (SAC) to create scan paths based on the provided saliency map. Finally, I assemble the pieces and show the results of the whole selective attention system working.

4.1 Experiments incorporating center-surround

One possibility to identify spatial discontinuities with a saliency-based attention system is to use the center-surround operation. This operation is implemented by a difference of center activity and surround activity as shown in Sec. 3.3.2. To calculate differences with neurons it has to be possible to stimulated them in both ways, excitatory and inhibitory, at the same time.

The first experiments described in this section were conducted to get a feeling for the IF2DWTA chip (compare Sec. 2.2.2) that carries out the mentioned operation. Therefore, the neuron's reaction to both the excitatory and the inhibitory synapses were examined independently.

In the second set of experiments both synapse types were stimulated to carry out the center-surround operation.

4.1.1 Stimulating inhibitory and excitatory synapses of a single neuron with computer generated spike trains

As described in Sec. 2.2.2, all neurons on the IF2DWTA have two excitatory and one inhibitory synapse that are accessible via AER. The following section shows how the neuron's output reacts on different input to both synapse types with different bias settings and different stimulus frequencies.

For these experiments only the IF2DWTA chip was used within the multi-chip setup. To obtain maximum control over the stimuli all stimulating events were generated on the workstation and transmitted via the AEX board to the neuromorphic chip. While the chip was stimulated the output was monitored. Either the frequency of the stimulus to the excitatory or the inhibitory synapse was kept fixed while the frequency to the other synapse type was varied in steps. For each frequency several runs were conducted to collect data for statistics. The stimuli for each run lasted 5 s to ensure that the chip reached a steady state. From the 5.5 s recorded only 3 s in the middle of the recording were used to calculate mean frequencies. This

method removes events from possibly occurring transient states at the beginning and the end of the recordings.

Investigating the excitatory synapses

To investigate the synapses of a neuromorphic chip one would usually stimulate the synapse and measure the resulting output current. The main purpose of the experiments presented here is to find appropriate bias values for the synapses and to show the relationship of a stimulus to the neuron's output firing frequency. Hence, in the following experiments the neuron's output was recorded while either of the synapses was under investigation. The biases for the neurons were default values used also for other experiments in the institute.

First, the excitatory synapses are investigated. Figure 4.1 shows the neuron's response to stimuli at different frequencies with different weight biases and different synaptic time constants for two typical neurons. As expected the higher the stimulating frequency the higher the neuron's output frequency. The same is true for increasing weight biases. With the synaptic time constant one can control the time the synapse integrates incoming spikes.

Depending on the weight parameter values of the excitatory synapse the neuron responses to the stimuli in different ways: If the weights are low the input-output relationship is almost linear as is shown by the red and blue traces in Fig. 4.1(a) and 4.1(c). Increasing the weight results in a more exponential response of the neuron for low input frequencies that ends in the neuron's saturation at about 1000 Hz (green trace). By increasing the weight even more the neuron behaves as a switch: as soon as there is input the neuron fires at its maximum output frequency (black trace).

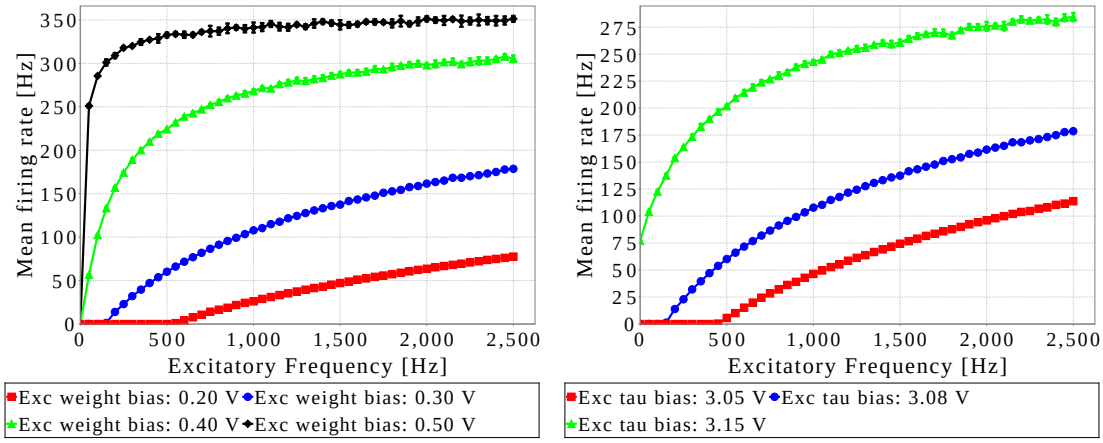
To use the IF2DWTA for center-surround operations a linear relationship between the input stimulus and the output frequency is desirable. This results from the requirement that different strength in stimuli should be represented in different input currents to the neurons.

With the synaptic time constant the range of stimulus input frequencies can be controlled. When the bias value is too small, i.e. the time constant is very short, the neuron is only responding to very high frequencies. When the bias value is too big, i.e. the time constant is too long, the small currents within the synaptic circuit are already big enough to make the neuron fire (green line in plots 4.1(b) and 4.1(d)). Although no input is provided the neuron keeps firing.

Investigating the inhibitory synapses

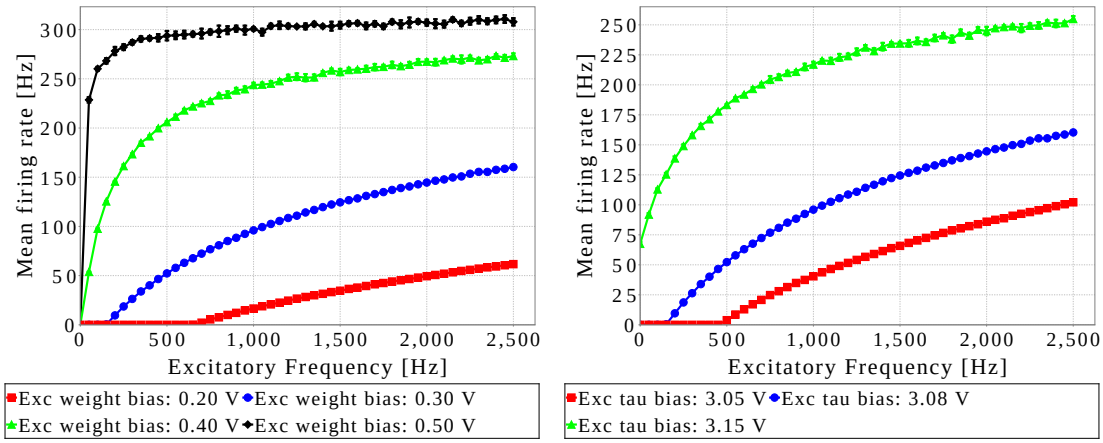
Since stimulating the inhibitory synapse reduces the activity of a neuron the neuron has to be excited while investigating the behavior of the inhibitory synapses. To create comparable results from experiment to experiment the level of excitement has to be the same. To provide the necessary input current to excite the neuron the IF2DWTA chip provides two possibilities: either the excitatory synapses are stimulated via AER or a current is directly injected into the neurons via a current source controlled by the injector current bias.

Preparatory experiments show that the same level of excitement can either be achieved by stimulating the neuron with a spike train of 2500 Hz with a weight bias of 0.2 V (compare red lines in Fig. 4.1(a) and 4.1(c)) or to set the injector current bias to 2.80 V. Figure 4.2 compares



(a) Cell 0,0 - Weights

(b) Cell 0,0 - Time constant



(c) Cell 25,4 - Weight

(d) Cell 25,4 - Time constant

Figure 4.1: Exploration of the excitatory synapse of the IF2DWTA chip. Curves show the neuron's mean output frequency when stimulated for 5 s at its excitatory synapse with a Poisson spike-train generated on a workstation. To avoid on- and off-set artifacts only the middle 3 s were used for the mean frequency calculation. Two parameters were altered: in Fig. (a) and Fig. (c) the weight parameter was changed whereas in Fig. (b) and Fig. (d) the synapse's time constant was varied. The bias voltage that controls the synaptic time constant was set to 3.08 V in the weight plots and the bias voltage that controls the synaptic weight was set to 0.30 V in the time constant plots. Hence the blue traces are pairwise the same for the same cell. The plots show the traces of two typical cells.

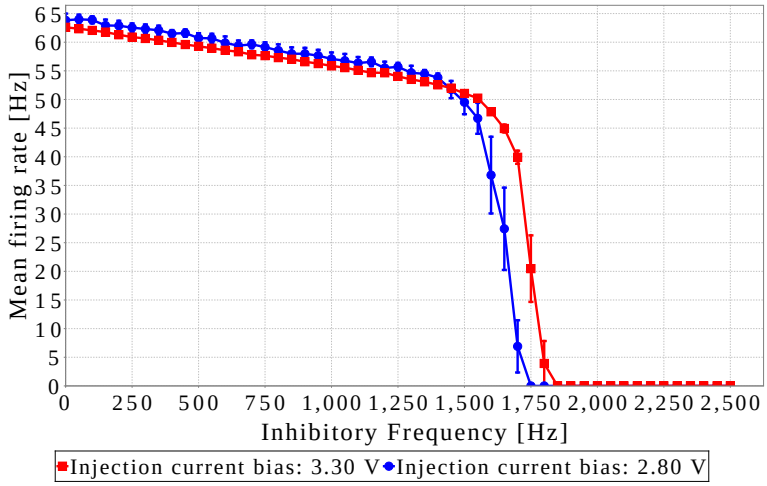


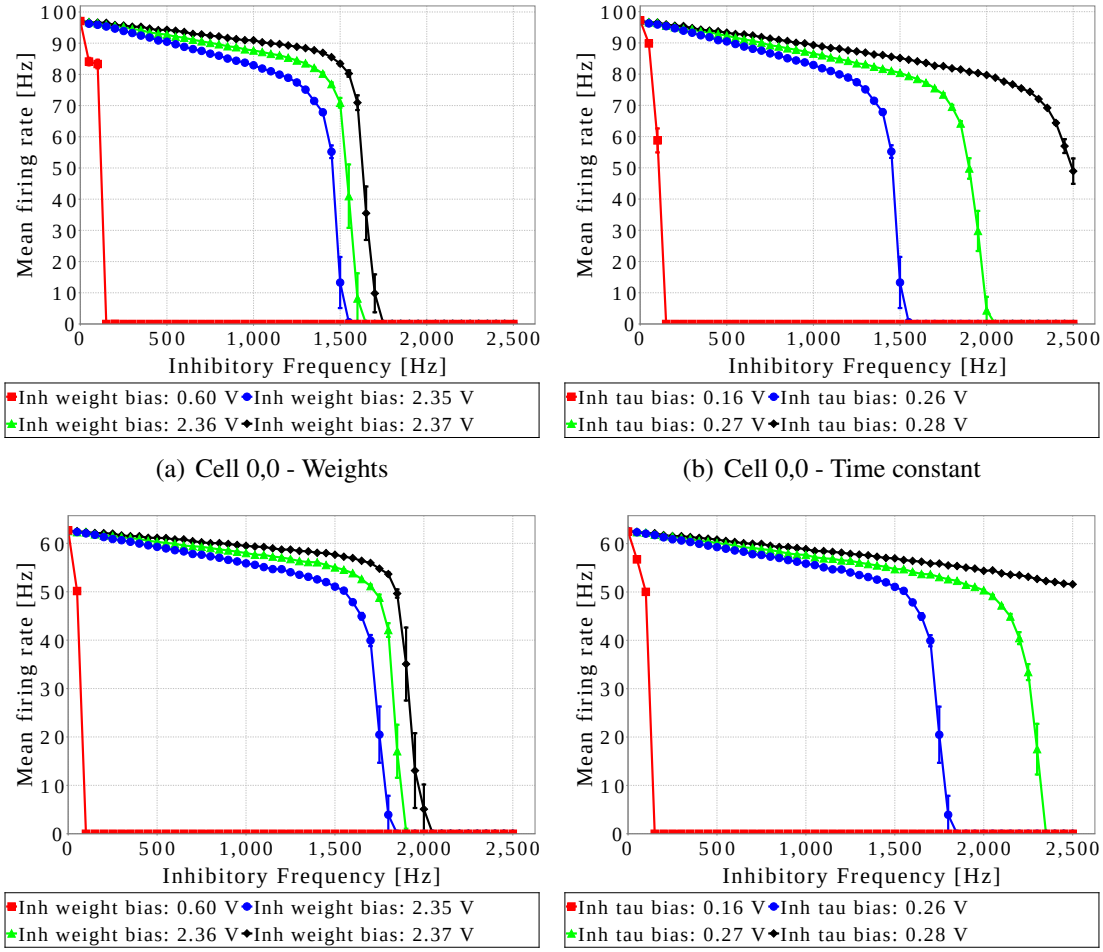
Figure 4.2: Comparison of two methods of the neuron's excitation. To investigate the neuron's inhibitory synapse the neuron itself has to be excited. In the IF2DWTA this can either be accomplished by an injector current bias or by the stimulation of the neuron's excitatory synapse via AER. In the plot the blue line was generated by exciting the neuron with the injector current whereas the neuron was excited by the external AER stimulus to create the red line. For this plot cell 0,0 was investigated.

example curves of both methods of excitement. Even though the results of the preparatory experiments suggest a similar behavior of the neuron by being excited by either method the latter method uses less circuit elements, i.e. the excitatory synapse. On the other hand stimulating the neuron via AER is closer to the use case of calculating the center-surround operations. Therefore, exciting the neuron by using the excitatory synapses via AER was further used in the presented experiments.

The expectation is that the higher the inhibitory stimulus the lower the neuron's output frequency. Experiments showed this works best when the neuron fires at a much lower rate than the frequency of the input stimulus. This results from the need that inhibitory spikes are only effectual when they are received during the neuron's integration time. If they are received just after the neuron spiked, i.e. during the refractory period, no current generated by the inhibitory spike can be subtracted from the neuron's membrane potential. This is why relatively small weights were chosen for the excitatory synapse experiments.

The results in Fig. 4.3 show that the general desired behavior, the higher the inhibitory stimulus the less the output frequency, can be accomplished with the IF2DWTA chip.

The experiments in Fig. 4.3 show three phases: In the first phase one can observe a linear relationship between inhibitory input frequency and the neuron's output frequency. In this phase the current created by the inhibitory synapse is much smaller than the excitatory current. The second phase is characterized by a steep decline of the neuron's output frequency. This is due to an exponential increase of the current created by the inhibitory synapse. Therefore, the neuron's output frequency decreases quickly. As can be seen from the much bigger values for the mean frequency's standard deviation in this phase the chip works in a unstable regime. In the third phase the incoming inhibition is so strong that the neuron's output frequency is very



(a) Cell 0,0 - Weights

(b) Cell 0,0 - Time constant

(c) Cell 25,4 - Weight

(d) Cell 25,4 - Time constant

Figure 4.3: Exploration of the inhibitory synapse of the IF2DWTA chip. Curves show the neuron's mean output frequency when stimulated for 5 s at its inhibitory synapse with a Poisson spike-train generated on a workstation. At the same time the neuron was excited via the excitatory AER synapse (excitatory weight bias at 0.2 V time constant at 3.1 V excitatory frequency of 2500 Hz). To avoid on- and off-set artifacts only the middle 3 s were used for the mean frequency calculation. Just as in Fig. 4.1 the inhibitory synapse's weight (Fig. (a) and Fig. (c)) and its time constant (Fig. (b) and Fig. (d)) were varied. The bias voltage that controls the synaptic time constant was set to 0.26 V in the weight plots and the bias that controls the synaptic weight was set to 2.35 V in the time constant plots. Hence the blue traces are pairwise the same for the same cell.The plots show the traces of two typical cells.

low or even zero. The inhibitory current is now much bigger than the excitatory one. Hence, the neuron is silent.

In the last sections I showed that the IF2DWTA's excitatory and inhibitory synapses can be operated in different regimes. Note that these results base on Poisson spike trains that are generated on the workstation.

4.1.2 Carrying out center-surround operation with stimuli provided by the Dynamic Vision Sensor

Based on the experience gathered during the conduction of the experiments described in Sec. 4.1.1 in the following section the IF2DWTA will carry out center-surround operations on data provided by the DVS.

The DVS is oriented towards a monitor where the stimulus is presented. A simple Python script based on the VisionEgg library [Straw, 2008] generates the visual stimuli. It is able to show up to three black colored targets either disks or annuli on a white background. Since the DVS is only able to detect changes in a visual scene (see Sec. 2.2.1) it is necessary to either move the targets or let them blink. While moving a target the DVS only generates events at the target's edges; by letting a target blink events are generated for its whole area. Therefore, the latter method is used in the experiments presented here. The blinking frequency can be controlled for every target independently. Because the higher the target's frequency the more events are generated by the DVS different target frequencies generate stimuli at different strengths. Also the size of the disks and both the inner and the outer diameter of the annuli, respectively, can be controlled. To control the different frequencies and diameters the script provides an interface accessible via a network connection. This allows to write programs controlling the neuromorphic system and conducting the experiments in any programming language – provided it supports establishing network connections – independently of the presenting Python script. The script is based on a library that already proved its use in other laboratories and was already used for visual physiological research (see citation page of VisionEgg's homepage[1]).

The DVS injects its data via an AEX board into the multi-chip system. As described in Sec. 2.1.4 the mapping device is needed to send events from the DVS to the IF2DWTA that will perform the center-surround operations. The mapping ensures two functions: First, it translates the addresses of events coming from the DVS into addresses of the IF2DWTA's synapses. Second, the mapping device is able to remove events from the incoming stream of events based on user controlled probabilities. This results in a weighing of the different input streams to the input synapses of the IF2DWTA. By this mapping the characteristic weight function of the center-surround operation, as described in Sec. 3.3.2, is realized.

Different approaches to implement the weight function

There are two possible approaches of implementing the weight function 3.1:

[1] http://www.visionegg.org

One way is to calculate for a every DVS-pixel with radius $r = \sqrt{x^2 + y^2}$ within a given radius r_s its weight $W(r)$. Is the weight $W(r)$ positive a mapping is established from the DVS-pixel to the excitatory synapse of the IF2DWTA with a probability $p = \frac{W(r)}{W_{max}(r)}$. W_{max} is set to 127 because the mapping probabilities are encoded with 7 bit (compare Sec. 2.1.3) and a probability of 1 corresponds to 127. Is the weight negative events from the DVS are mapped to the neuron's inhibitory synapse with weight $|W(r)|$. Since the DVS' pixel creates events if the light intensity increases and if the decreases – distinguishable by bit 1 in the address – there are two mappings per pixel from the DVS to one neuron on the IF2DWTA.

The other possibility is to implement the two Gaussians independently and let the neuron calculate the sum. This is accomplished by establishing a mapping from a DVS-pixel to the neuron's excitatory synapse on the IF2DWTA if the first term of the weight function 3.1, $k_c \exp\left[-\left(\frac{r}{r_c}\right)^2\right]$, is bigger than zero. If the second term, $k_s \exp\left[-\left(\frac{r}{r_s}\right)^2\right]$, is also bigger than zero the DVS-pixel has to be linked with a mapping to the neuron's inhibitory synapse. By this approach up to four mappings per DVS-pixel to one neuron have to be established. With this approach the neuron is used to calculate the difference of the weight function $W(r)$.

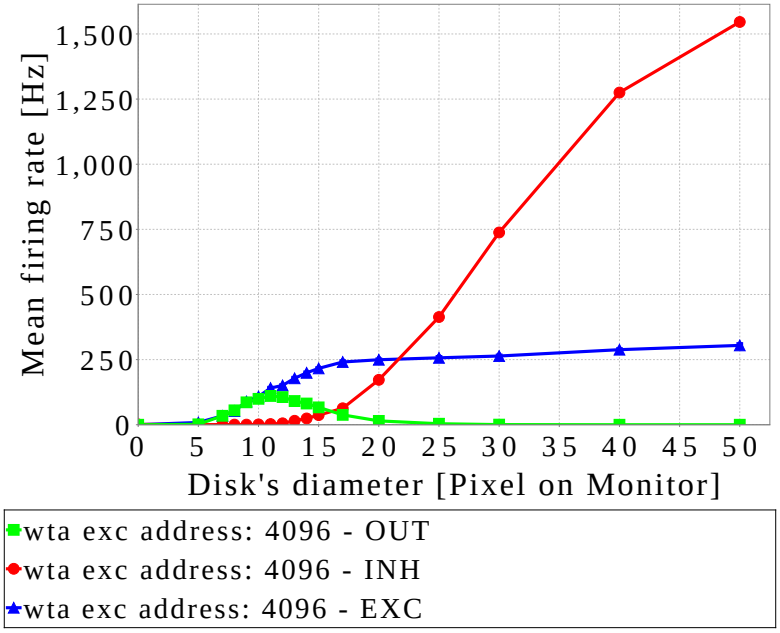
In both approaches one neuron on the IF2DWTA receives input in form of currents generated by its synapses from several DVS-pixels according to the weight function 3.1. These currents are summed up and determine the level of the neuron's membrane potential. If the sum is big enough one or more spikes are generated. The output frequency represents therefore the weighted input from the DVS pixels: it executes the center-surround operation.

When the second approach is used the neuron has to execute both, the summation over the inputs of DVS-pixels as well as the difference of excitation and inhibition for each pixel. This additional calculation increases the complexity of the operation the neuron has to carry out. A less complex system is usually easier to control hence the first approach was implemented to conduct the following experiments.

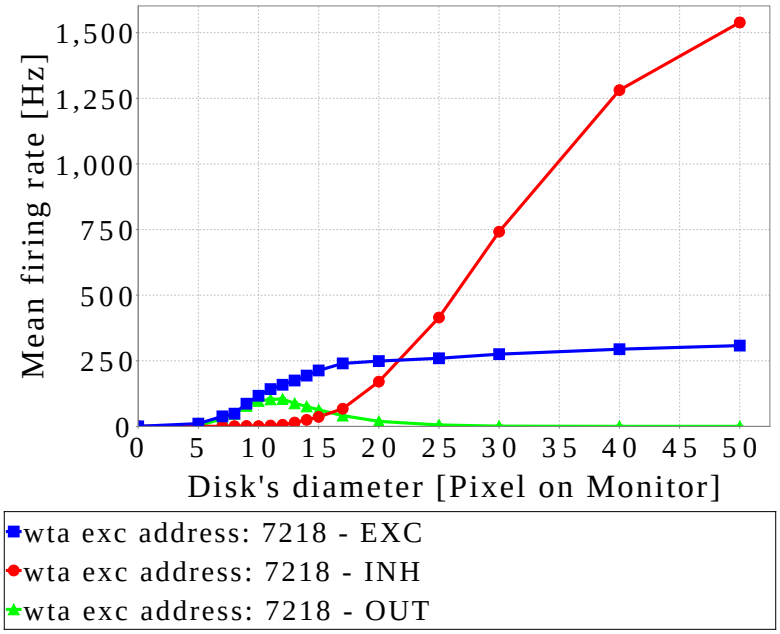
"On"-cell experiments

"On"-cells are cells that respond most to a visual stimulus within their receptive field that has a small spot of high activity within an area with low activity (compare Fig. 3.4, III). To examine the behavior of the IF2DWTA to such a stimulus black blinking disks with different diameters were presented to the DVS. The diameter's values expressed in pixel on the presenting monitor range from 0 till 50 pixel. The upper value was chosen such that the disk's size exceeds the neuron's receptive field. The blinking frequency was kept constant over all experiments at 5 Hz. The vision sensor generated events were mapped to the input synapses of the IF2DWTA. For testing purposes only one cell of the chip at a time was picked to which the mapping was established. The stimulus was presented to the DVS via a monitor for 5 s. Both, the stimulus events and the neuron's response were recorded by a workstation. From the recorded data only the 3 s in the middle were taken for further investigation. This was done to avoid stimulus on- and offset artifacts.

Figure 4.4 shows the results of experiments with an "on"-cell mapping. In this figure, both, the mean frequencies of the inputs to both synapse types and the mean frequency of the output of the neuron carrying out the center-surround operation are shown: If the disk is very small



(a) Cell 0,0



(b) Cell 25,4

Figure 4.4: "On"-cell experiments: A blinking disk with different diameters shown on a monitor is recorded by the DVS. The disk's diameters are shown on the x-axis in number of pixel on the monitor. The blue and the red trace show the mean frequencies of the mapped and weighted input events to a neuron on the IF2DWTA. The blue trace shows events sent to the neuron's excitatory synapse whereas the red trace shows the events sent to its inhibitory synapse, respectively. The green trace represents the neuron's response to the input. Shown are the results for two typical cells.

the events sent to the neuron do not arouse any output. Increasing the disk's size the neuron's excitatory input (blue trace) increases quicker than the inhibitory input. This is due to the weighing function 3.1: The Gaussian of the center region is very narrow and its maximum value is much higher than the Gaussian of the surround which is wider but not so strong. In the case of the "on"-cell the narrow Gaussian is connected to the excitatory synapse whereas the wide Gaussian is connected to the inhibitory synapse. At a disk size of 11 pixel on the monitor the neuron's output activity peaks. This is the case when the disk size corresponds to the zero crossing of the weight function. Then, all events of the DVS are sent to the excitatory synapse. As soon as the disk size increases further events from the DVS are also sent to the neuron's inhibitory synapse. This decreases the neuron's activity until the neuron becomes silent.

To obtain this behavior the biases for the excitatory synapses were set to 0.4 V for the weight and 3.08 V for the time constant, respectively. These values correspond to the green trace in Fig. 4.1(a) and 4.1(c). Even though the accumulated inhibitory input to the neuron is much larger than the excitatory one the inhibitory synapses had to been set to very extreme values to turn the neuron's activity off reliably: 0.16 V for the time constant and 0.6 V for the weight bias. The red curves in Fig. 4.3 give an impression of the inhibitory strength obtained by using theses values.

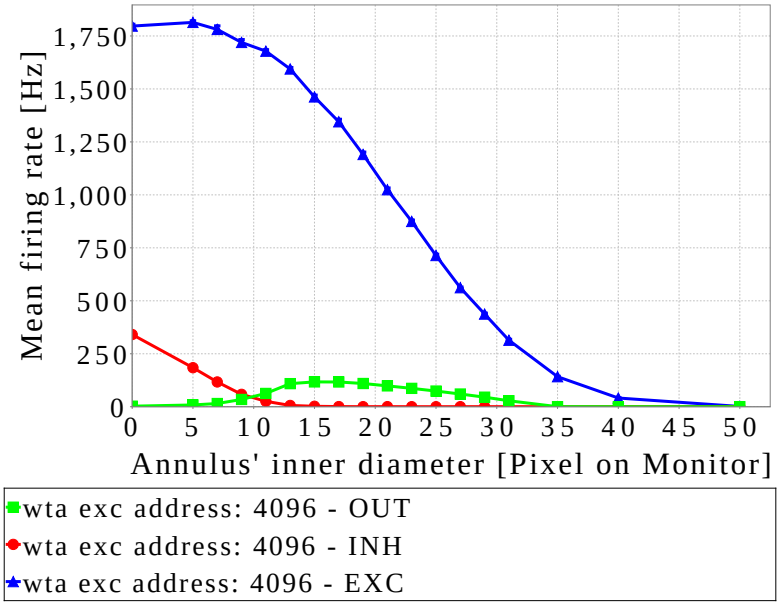
The results obtained by these experiments show that is possible to implement a "on"-cell center-surround operation with the IF2DWTA: The neuron shows no output if there is no input; the neuron reacts to a small high active visual stimulus most and its activity decreases with increasing stimulus size. If the stimulus cross a certain size the neuron generates no events anymore.

"Off"-cell experiments

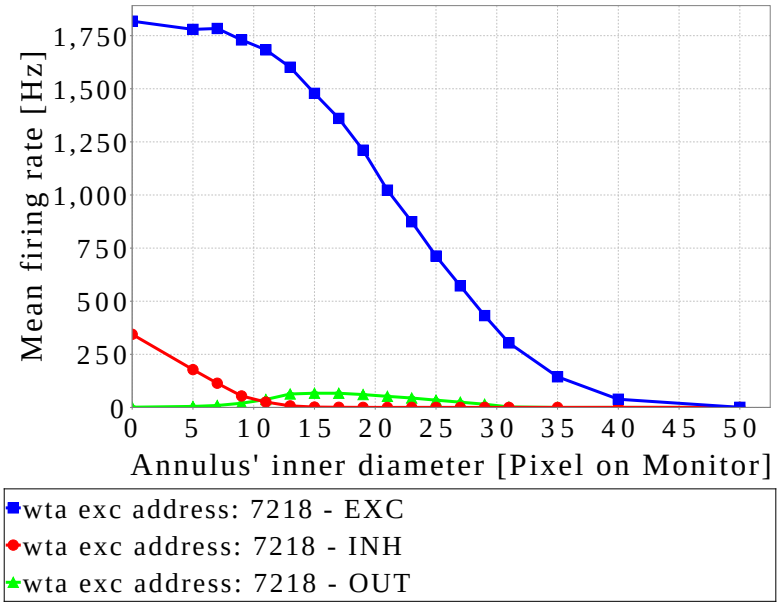
In contrast to the "on"-cell the "off"-cell's preferred stimulus is a low active spot within a high active environment (compare Fig. 3.4 IV). But just as the "on"-cells "off"-cells should not be active if there is no input or if there is only a big high active stimulus presented.

The experimental setup for the "off"-cell is the same as for its counterpart but the presented stimulus differs: Instead of a disk an annulus with different inner diameters is used. The inner diameters vary from 0 up to 50 pixel on the monitor. The outer diameter is kept fixed at 75 pixel which is bigger than the cell's receptive field. The stimulus is presented for 5 s but only the middle 3 s of the recordings are used for further investigation.

Figure 4.5 summarizes the findings for the "off"-cell experiments: If the inner diameter is of zero size a big disk is presented to the DVS. This corresponds to the extreme case II in Fig. 3.4. According to the requirements of the center-surround operation this stimulus should not generate any output. As can be seen from the Fig. 4.5 both the excitatory (blue traces) and the inhibitory (red traces) input are at their maximum. To suppress the excitatory input the inhibition has to be strong enough especially because the inhibitory input is much lower than the excitatory one. By increasing the size of the annulus' inner diameter the part of the stimulus that generates the inhibitory input diminish. This is reflected in the earlier drop of the red trace in Fig. 4.5. As inhibition decreases the excitatory input is more and more able to activate the neuron's output. The output frequency peaks when the inhibitory input is zero.

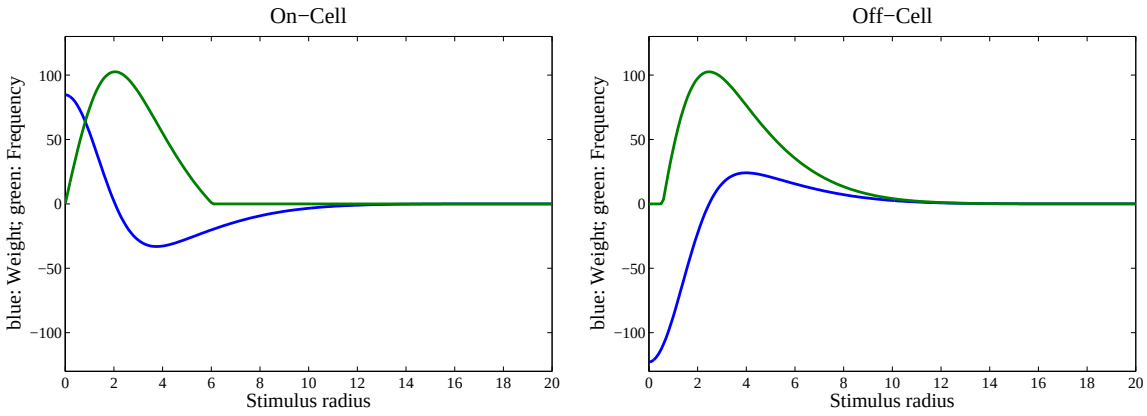


(a) Cell 0,0



(b) Cell 25,4

Figure 4.5: "Off"-cell experiments: A blinking annulus with different inner diameters shown on a monitor is recorded by the DVS. The outer diameter is kept at 75 pixel. The annulus' inner diameters are shown on the x-axis in number of pixel on the monitor. The blue and the red trace show the mean frequencies of the mapped and weighted input events to a neuron on the IF2DWTA. The blue trace shows events sent to the neuron's excitatory synapse whereas the red trace shows the events sent to its inhibitory synapse, respectively. The green trace represents the neuron's response to the input. Shown are the results for two typical cells.



(a) On-Cell (b) Off-Cell

Figure 4.6: Simulating the "on"- and "off"-cell responses. The blue curve represents the input weight function. It is derived from the weight function 3.1. The neuron's response is shown by the green curve. On the x-axis the stimulus' radii is represented whereas the y-axis is used for both, the weight and the output frequency. All units are arbitrary. The figure shows the differences of the cell output in response to their preferred stimulus at different sizes.

This corresponds to an annulus' inner diameter of 14 pixel. Increasing the blank no activity generating spot of the stimulus further also decreases the excitatory input. This reduces also the neuron's output frequency.

Because the "off"-cell mapping generates a much higher excitatory input the excitatory weight bias is set to a lower value than for the "on"-cell case: 0.3 V (compare blue traces in Fig. 4.1(a) and 4.1(c)). The bias value used for the time constant is the same (3.08 V). Since the IF2DWTA cells only provide one inhibitory synapse per cell both cell types share the same inhibitory bias values (bias(w)=0.6 V, bias(τ)=0.16 V).

In a nutshell, it is possible to implement the "off"-cell operation as with the "on"-cell center-surround operation. The results show that the neuron does not generate any output if there is only a big high active stimulus. The output frequency increases with the size of a silent center region until it peaks at a characteristic size. Further increasing the inner diameter decreases the neuron's output frequency.

Comparison of "on"- and "off"-cell responses

One notice by comparing Fig. 4.4 and 4.5 that the output of the "on"- and the "off"-cell differ. First, they peak at different stimulus sizes: 11 pixel in the "on" case and 15 pixel in the "off" case. Second, the range of input stimuli where the output neuron generates events is wider in the "off"-cell case than in the "on"-cell case. And third, to obtain approximately similar maximum output frequencies for better comparison the excitatory weight biases were chosen different: 0.4 V for the "on"-cell and 0.3 V for the "off"-cell, respectively.

I simulated the experiments carried out briefly with a Matlab script to try to find an explanation for these differences in the neuron's output. The underlying neuron model is the one

described by Eqn. 3.2 with $C = 1$. Hence, the model's input is the integral of the weight function 3.1 with the limits given by the stimulus size. In case of the "on"-cell the integration starts at zero until r because for a given r this is the region that generates input events. In contrast, for the "off"-cell the integration limits are from r until the outer stimulus radius (set to 20 in the presented simulation) since in the annulus case the outer region is active and generates the input. I simulate the different stimulus sizes by varying the integration limits. Exactly the same parameter values for k_c, k_s, r_c, and r_s were used for the weight function in the simulation as for the experiments with the multi-chip setup. To model the different bias values for the excitatory synapses for the different cell types and the different weights for inhibitory and excitatory synapses constants were introduced into the calculation of the weight function. The difference between the excitatory weights was assumed to be 10 % and the inhibitory synapses were assumed to be 40 % stronger than the weaker excitatory synapse. It is very difficult to compare the strength of the synapses within the context of the neuromorphic system therefore the difference of the synapse types was chosen arbitrarily but in a plausible range. A higher value reduces the output frequency of both and shifts the peaks of the "on"-cell the "off"-cell further apart. The 10 % difference was chosen such that the output frequency of the "on"- and the "off"-cell is approximately the same.

Figure 4.6 shows the result of the simulation. The green graphs show the neuronal output whereas the blue graphs show the input. Comparing the location of the output peaks show that for the "on"-cell it is earlier (at $r = 2$) whereas the "off"-cell's output peaks at $r = 2.5$. This is in accordance to the different preferred stimulus sizes in the experiments. Another accordance can be observed by looking at the width of the output curves: the "on"-cell fires for stimuli of size 0.1 until 5.9 ($\Delta = 5.8$) whereas the "off"-cell fires for a much greater variety of stimuli (0.6 until 12.0, $\Delta = 11.4$). Output was considered present when the value was greater than 1% of the output's maximum. Note that the metrics used in this simulation cannot directly be compared with the metrics used within the context of the experiments.

By examining this simplified simulation the differences of preferred stimulus size and the width of the active curves can be traced back to the use of the same mapping parameters for both, the "on"- and the "off"-cells.

Conclusion

The presented results show that it is possible to carry out the center-surround operation on the IF2DWTA by carefully choosing appropriate bias values and mappings. I will use this operation to identify spatial discontinuities in the context of the saliency-based attention model proposed by Itti et al. [1998] later in this thesis. Even though the parameter values, especially for the weight function, were derived from experiments done with the cat's retina it is clear that attention is not calculated in the retina or some other brain areas as the LGN. I use these data only to derive a model for the center-surround operation and use it in the context of attention without claiming any direct relationship.

With different bias sets different behaviors of the excitatory synapses are possible: The relationship between input and output frequency can either be linear or exponential. In extreme cases the neurons behave almost like a switch: Whenever any input is sent to the excitatory synapse the neuron starts firing. Stimulating the inhibitory synapses the output of an excited

neuron shows only two behaviors: Either a slow decay of the output frequency when the inhibitory input rises followed by a cut-off or a switching behavior: Whenever an inhibitory input is provided the excited neuron stops firing. Calculating e.g. the differences of maximum output firing rates that the neurons show values between 14 and 40 % can be observed. These differences arise due to fabrication mismatch. Nevertheless, the general behavior of the neurons across the chip is reliable (with one exception). Several circuit blocks are involved: Both, excitatory and inhibitory synapses and the neurons. All these circuits consist of several transistors so that there are several sources for the mismatch observed. Taking advantage of the probabilistic mapping device is a way to balance these differences. This method allows to correct both the input as well as the output frequencies. The same stimulus is sent to the synapses of different neurons. The output is recorded. Because the probabilistic mapping device is only able to remove events from a spike train all output frequencies have to be adopted to the lowest one. The probability p for each mapping from neuron i is calculated by:

$$p_i = \frac{f_{min}}{f_i}$$

f_{min} represents the lowest output frequency and f_i the output frequency of neuron i. This normalization method is not used in any experiments presented in this thesis.

4.2 Testing the generation of scan paths

In the following section the SAC chip's ability to generate scan paths is investigated. Therefore two sets of experiments were carried out: In a first set of experiments we measured the response of the SAC to different stimulus conditions without activating the pan-tilt-unit motors. In a second set of experiments we activated the control loop and used the events produced by the SAC to orient the vision sensors. Major parts of Sec. 4.2 were already published in Sonnleithner and Indiveri [2012].

4.2.1 Covert attention experiments

To examine the SAC's response to different visual inputs, we stimulated the DVS by presenting different patterns on an LCD screen, and analyzed the SAC output address-events. The DVS was stimulated by three blinking black rectangles on a white background of the LCD screen. We used blinking frequencies ranging from 5 to 30 Hz. The size of the rectangles was chosen such that in most of the cases only one pixel in the SAC was stimulated.

Due to the "real-world" conditions used in this experiment, namely the refresh rates of the LCD screen, the mapping of the 128×128 DVS pixels to the 32×32 SAC pixels, the variability in the illumination conditions, and the mismatch and inhomogeneous properties of both DVS and SAC VLSI circuits, the spike-trains received by each SAC pixel do not have a regular 5 to 30 Hz frequency. Rather, they are inhomogeneous, with periods of bursting activity interleaved by periods of noisy low frequency. The inter-burst frequencies are proportional to the visual stimuli blinking frequencies.

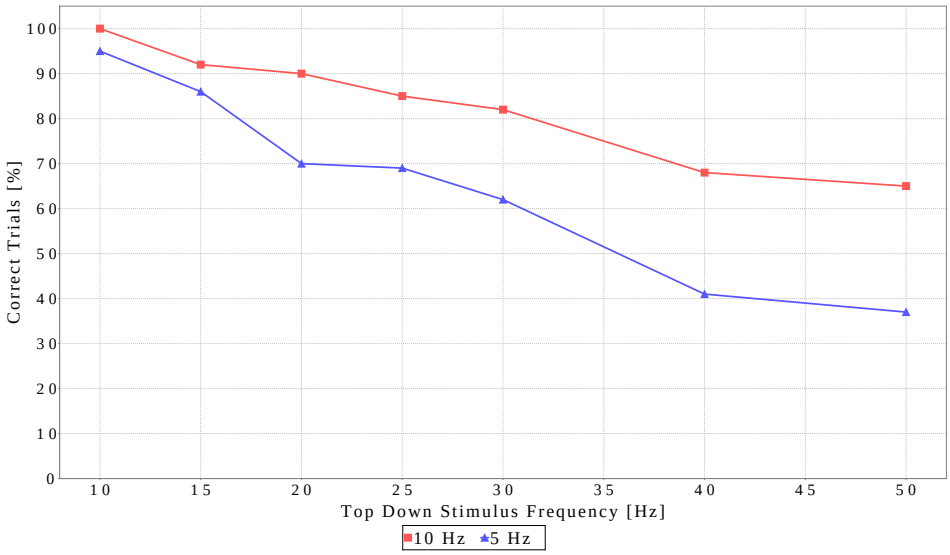


Figure 4.7: Percentage of correct trials for different distractor frequencies. The X-axis represents the top-down stimulus frequencies. In a correct trial, the SAC reports the location of the distractor rather than the top-down stimulus location. The Y-axis shows the percentage of correct trials.

This experimental setup was chosen as a compromise between "natural" scene stimuli (that would be used in typical operating conditions), and well controlled stimuli (e.g. produced by function generators or computers), in order to determine the system's settings, for optimal operation in natural conditions, while having good control of the stimulus properties.

In these control experiments the SAC is expected to detect the rectangle that blinks with the highest frequency (i.e. the salient target) and ignore the two distractors blinking with a lower common baseline frequency. As done in psycho-physics experiments, we set parameters in our experiments at threshold, so that the system would not select the right target 100 % of the times, and measured the equivalent of psychometric curves on the artificial system, by gradually increasing the difference between baseline stimulus frequencies and target stimulus frequencies. We ran two sets of experiments with different baseline frequencies: one with 5 Hz, and the other with 10 Hz (see Fig. 4.8). Furthermore we repeated the experiments with an additional input generated synthetically on the workstation, as a sequence of extra address-events merged to the stream of address-events coming from the sensor, to apply the concept of top-down attention to the system.

Experiment description

Each experiment comprises three 5 s lasting runs. Before the beginning of each experiment run, the system was reset to an initial state: the weights of the input excitatory synapses were set to zero, the WTA circuit bias current was turned off and the leak of the output neurons was set to max. At the onset of each run these parameters were reset to their default values.

To account for mismatch effects from both DVS and SAC circuits, we chose the locations

of the three black rectangles randomly for each experiment, but kept them fixed for each of the experiment's runs. During the three runs, the target was permuted among the three locations. We swept the target frequency from the baseline value (either 5 or 10 Hz) up to 30 Hz. Higher target frequencies could not be used, due to interference with the monitor or system refresh rate. For each target frequency chosen, we repeated multiple trials of the experiments and calculated the percentage of correct choices made by the SAC. To estimate how reliable the selection of the correct target is, we repeated the same set of experiments, using the same randomly picked locations, multiple times (see error-bars in Fig. 4.8).

As a next step, we set appropriate weights to the inhibitory synapses to activate the IOR mechanism in the winning WTA cell. This feature should allow the system to scan through all salient regions (i.e. the three blinking rectangles), but ideally the location of the strongest stimulus should be chosen more often than the distractors.

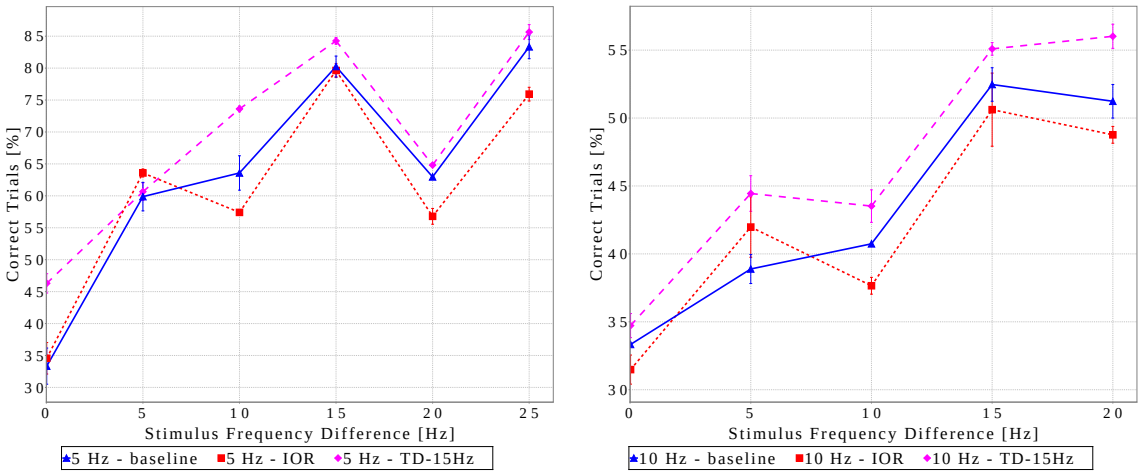
Finally, we were interested to test if the concept of "top-down attention" is applicable to our system and to see how it would influence the performance of the detection of the salient target. We simulated top-down influence by using a computer-generated stimulus that provides an additional input to the location of the target rectangle, and measured its effect on the selection process. The stimulus was chosen such that it would not always win the competition process against the distractors, if presented in isolation (without the visual target). Therefore we generated an artificial 15 Hz Poisson spike train that stimulated an area of 3×3 SAC pixels centered at the location of the visual target and applied it in parallel to the visual "bottom-up" stimulus.

To calibrate the top-down stimulus in a way that it would not alter the bottom-up selection process if presented alone (i.e. to find the appropriate top-down stimulus frequency), we stimulated the SAC with the top-down Poisson spike train while displaying a visual stimulus corresponding to single rectangle blinking either at 5 Hz or at 10 Hz at a different spatial position, and evaluated the competition process. Then we counted the number of times the bottom-up visual stimulus was selected and related it to the total number of trials. The results of these control experiments are shown in Fig. 4.7. Since there is a significant drop at 20 Hz top-down stimulus frequency, we chose maximum frequency of 15 Hz for the top-down spike-train.

Results

For each experiment run, we recorded both the input events mapped to the SAC and the SAC output events. For each target-distractor frequency pair we counted the runs where the SAC chose the target stimulus correctly and related it to the total conducted runs. The percentage correct results are summarized in Fig. 4.8. As expected, when all three stimulus rectangles are blinking at the same (baseline) frequency the system picks one location at random (33 % correct trials). This happens for both sets of experiments, with different baseline frequencies. As the difference between the target and the distractor frequencies increases, the percentage of correct runs increases. The drops in performance at 20 Hz for the 5 Hz baseline frequency correspond to an absolute target frequency of 25 Hz. Therefore it is most likely due to artifacts induced by interference with the power line or the screen's refresh rate.

When activating the SAC's IOR mechanism, the system's performance is less regular. This



(a) 5 Hz distractor frequency

(b) 10 Hz distractor frequency

Figure 4.8: Percentage of correct trials for different baseline distractor and target stimulus frequencies. The X-axis represents the *difference* between the distractor baseline frequency and the target blinking frequency. The Y-axis represents the percentage of correct trials. The dotted lines report the results of experiments with the IOR mechanism activated. Dashed lines show the results obtained with the additional top-down input. Error bars represent the standard deviation. There is a drop in performance at 20 Hz for the 5 Hz distractor frequency experiments. As this corresponds to an absolute stimulus target frequency of 25 Hz, the drop in performance is most likely due to artifacts due to interference with by the power line or the screen's refresh rate.

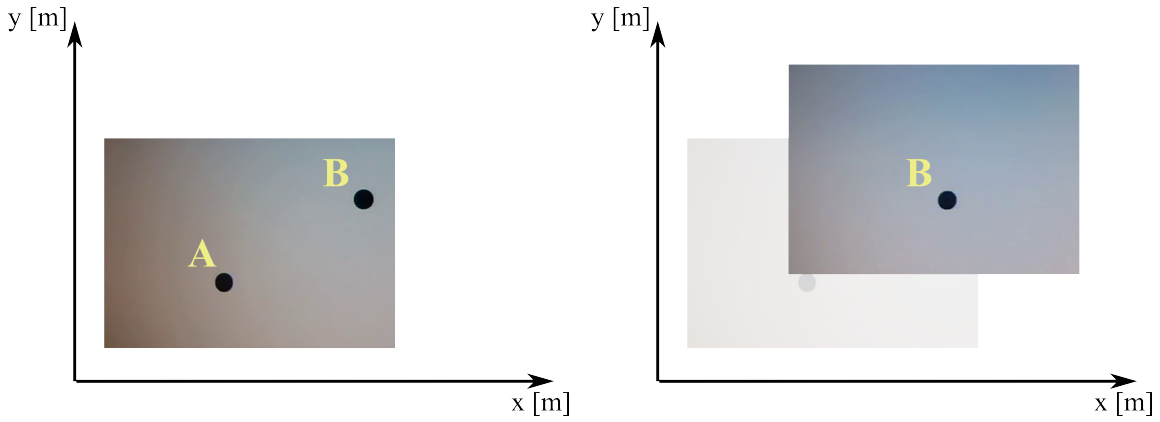
is expected since this mechanism introduces additional dynamics into the selection process.

As expected, the top-down stimulus can positively bias the selection process: the system's performance in choosing the correct rectangle increases for both baseline frequencies (see dashed lines in Fig. 4.8).

4.2.2 Overt attention experiments

In this section we describe experiments in which the active vision system orients the camera and the DVS toward salient regions. Specifically, we oriented the dynamic vision sensor towards a standard LCD screen and presented visual stimuli provided by a Java program that we developed for this purpose. The stimuli consisted of two blinking disks on two fixed locations A and B (see Fig. 4.9). We chose stimuli locations A and B such that they lay both in the DVS field of view, and such that both axes of the pan-tilt-unit had to move (pan: about 12°, tilt: about 8.5°) in order to shift the DVS to center location B in its field of view, from location A.

At the beginning of the experiment, a disk blinking at a frequency of 10 Hz was presented at location A, and the DVS was centered on A. After 5 s, a blinking disk of 20 Hz appeared at location B. At the same time, the disk at location A stopped blinking. After 5 s the blinking location was switched back, then blinking at a frequency of 30 Hz. The experiment ended



(a) Start position (b) End position

Figure 4.9: Overt attention control experiment: (a) while the system is focusing on the bottom left dot A, the top right dot B appears and starts to blink. The system selects the new input B as the winner and eventually it makes a saccadic camera movement to centers the new target in its field of view (b). The system uses the DVS to calculate the field of view center, and the stimuli A in (a) and B in (b) are not in the center of the color vision sensor images because it is not perfectly aligned with the DVS.

after another 5 s. The increased frequencies made sure that the newer stimuli were always more salient than the preceding ones.

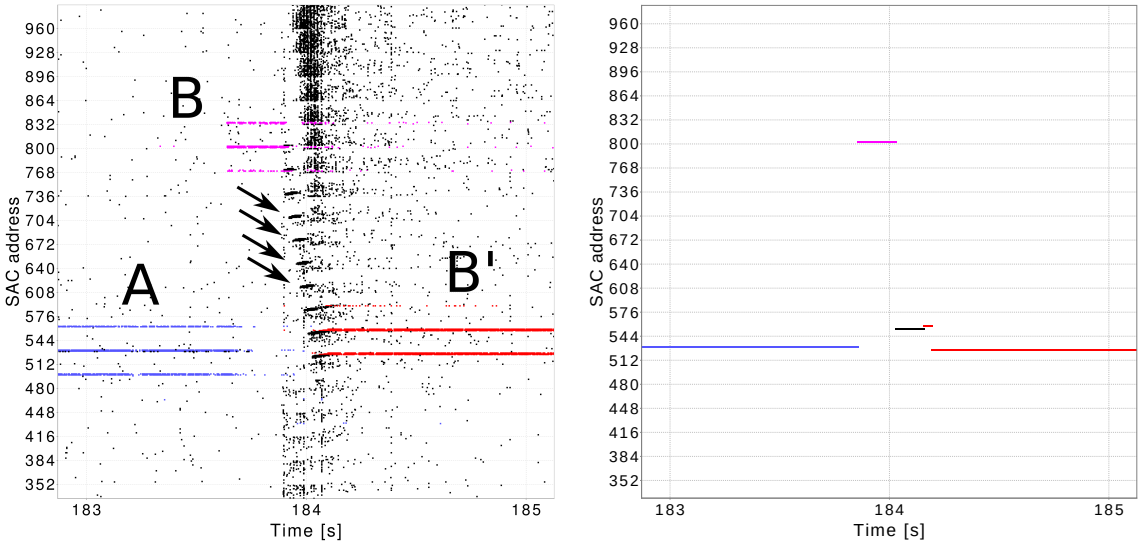
Both the stimulus data sent to the SAC and the output data produced by the SAC were recorded. Figure 4.10 shows an example of raw address-event data: The plot's horizontal axis shows the experiment's time in seconds. Each dot in the figure represents the occurrence of an event. To represent the two dimensional structure of the chip, the pixels' x- and y-coordinates were collapsed on the y axis ($pos = x + 32y$).

During this control experiment the SAC's IOR feature was not enabled.

Measurement

The raw address-event data was analyzed to measure the active vision system's reaction times. To get a better visual representation of the data, the addresses that represented the blinking disks were highlighted by colors (see Fig. 4.10). During the first phase (highlighted in blue), the system fixated the blinking disk at location A. At about 183.7 s the second disk at location B began to blink. In the raster plot, this phase is colored in pink. After a short time the system reacted on this new input and the pan-tilt-unit began to move. This phase can be easily identified by the high activity throughout all DVS addresses around 184 s. The arrows in Fig. 4.10(a) point to the clusters of spikes generated by the disk moving from B to B'. Finally, in the third phase of the experiment the system has centered the location B (colored in red, indicated with letter B').

On average, with the biologically plausible time-constants and settings used in these experiments, the system takes 128 ms ($\sigma = 25.3$ ms) to shift from one location to the next. As observed in the raster plot of Fig. 4.10(b), and as expected by the WTA operation of the



(a) SAC input spikes

(b) SAC output spikes

Figure 4.10: Raster plots of spikes representing the SAC input (a) and output (b). Each dot in the plot corresponds to an address-event. To represent the two dimensional structure of the chip, the pixels' X- and Y-coordinates were collapsed on the Y axis ($pos = X + 32Y$). Arrows indicate the clusters of spikes generated from the disk at location B during the camera movement.

SAC, there is only one winner at a time. After the winner is chosen, the system takes 28 ms ($\sigma = 1.4$ ms) to start a new saccadic camera movement (latency measured from the first output spike produced by the SAC). We used the significant increase in overall activity of the DVS to define the time of saccade onset. With the beginning of the onset of a saccadic camera movement we measure the final figure of merit: the time required by the pan-tilt-unit to center the new salient region in the DVS field of view. We define the end of such period by using the spikes produced by the SAC at the new location. For this time period the system requires 324 ms ($\sigma = 18.2$ ms).

The overall time used by the active vision system to select a new target and move the sensors to center it in its field of view can be obtained by summing up the time of these three different phases. This results in less than 500 ms. Both SAC and motion latencies can be easily decreased and tuned to the experiment/system requirements. In this experiment we purposely biased the SAC to have biologically plausible response properties, which result in these relatively high latencies.

4.2.3 Conclusion

The experiments carried out in this section show how the SAC can be used to pick from a stimulus, e.g. created by the DVS, the region with the highest activity. This computation can be influenced externally by a "top-down" signal. By using the built-in IOR mechanism it is possible to pick not only the region with the highest input signal but to identify several regions

with high activity in a serial fashion. This information can be used to control a pan-tilt-unit to align the source of high activity to the sensor's center.

The SAC can be used to identify quite reliably salient regions in its input. The detection ratio can be improved if an additional input is provided at a location that should be chosen. This fact is not surprising. The higher the input at a certain location the more current is generated by the excitatory synapses which makes it more likely that the WTA network will choose this particular location. Nevertheless, this proof-of-concept shows that it is technically possible to fuse events generated by the workstation with events from the DVS. The more intriguing question is the following: In a more advanced system with a visual and an auditory sensor could this method be used to fuse stimuli from different modalities? Assuming the auditory sensor can only distinguish if a sound comes from the left or the right side of the robot's head. Spike trains from sound sources from the right could then be mapped to the right half of the SAC. Events from the left are mapped to the left half, respectively. If a noise in the right hemisphere occurs the right half of the SAC is already biased. A visual event in the right hemisphere would then be discovered quicker than without the input from the auditory sensor. Testing this use case in mind was the reason for the series of experiments with the additional signal.

4.3 The neuromorphic selective attention system in action

In this section I put together the different pieces described in the previous sections to build a neuromorphic attention system. It consists of the visual sensor, the DVS, the IF2DWTA that performs the center-surround operation and the SAC to choose the regions with highest saliency and therefore the regions to attend to. All the experiments described in the previous sections shed light on the different building blocks that will now be put together to a complex multi-chip system. After describing the system in detail, I will present the results of experiments carried out.

4.3.1 The details of the neuromorphic attention system

The neuromorphic attention system consists of three neuromorphic devices: the DVS, described in Sec. 2.2.1, the IF2DWTA, where details are given in Sec. 2.2.2, and the in Sec. 2.2.3 presented SAC. Each of them has a well defined function in the system. Figure 4.11 shows an overview of the system and the experimental setup.

The visual input is provided by the DVS. It generates events whenever it sense a change in contrast in its visual field. These changes can be elicited by moving, appearing or flickering objects or light sources. The next processing step is to detect spatial discontinuities which are regions in the visual scene that stand out from their surrounding. According to the model of Itti et al. [1998] they can be detected by a center-surround operation. These operations are carried out by the general purpose neuromorphic chip IF2DWTA. Finally, the decision which of these spatial discontinuities is most strongly prominent in the visual scene is taken by the

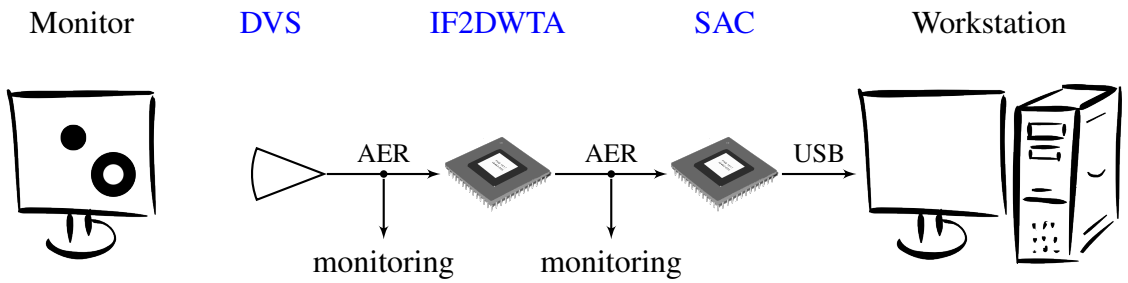


Figure 4.11: Schematic of the neuromorphic attention system setup. On a monitor different stimuli can be presented to the attention system. Stimulated, the DVS generates events that are weighted and transmitted to the IF2DWTA. There a center-surround operation is executed. Its result is sent to the SAC that chooses the location with the highest input rate. This location is – encoded by events – forwarded to the workstation. It is possible to monitor the different event streams to oversee the whole system.

SAC. To not only determine one salient location this chip has a built-in mechanism, called IOR, to switch to the second most, and – depending on parameters – n^{th}-most salient locations.

The system's logical topology is purely feed-forward, as shown in Fig. 4.11. Nevertheless, it is implemented by the ring structure described in Sec. 2.1.4. Figure 4.12 shows schematically the mapping of the system. The goal is to create overlapping receptive fields on the DVS that send events to the "on"- and "off"-cells. As shown in Sec. 3.3.2 for each receptive field there has to be both cell types. Hence, the number of neurons provided by the IF2DWTA has to be divided by two. The chip provides a sheet of 64×16 neurons with two excitatory and one inhibitory synapses. Hence, it is possible to create 512 pairs of "on"- and "off"-cells. Unfortunately this number does not permit to create a quadratic array of cells so that the closest values, 23×22, was chosen as the "attentional space" on the SAC. This means that only a central region with 23×22 pixel of the SAC gets an input from the IF2DWTA. Even though 518 cells are unused on the SAC the loss on the IF2DWTA is only six. The possibility of using a 16×32 sized "attentional space" was withdrawn: First, the aspect ratio is either very wide or narrow which stands in contrast to the aspect ratios of both, the DVS and the SAC. Also one could argue that our eye's aspect ratio is more quadratic than wide. Second, one of the sides would then use up the whole length of the SAC and therefore, assuming a retinotopic mapping, the receptive fields on the DVS would exceed its available range.

The center-surround mapping uses the mapper's probability mapping feature (see Sec. 2.1.3) to create a weighted connection from the DVS to the synapses of the IF2DWTA following Eqn. 3.1. The parameters were set to $r_c/r_s = 1/3$ and $k_c/k_s = 3$. The radii's ratio matches values found in biology whereas the second parameter ratio was chosen such that it meets the conditions found in Sec. 3.3.2. In my implementation, in contrast to the other values, the value for r_c can be chosen at run time, i.e. this parameter can be varied easily from experiment to experiment. Nevertheless, for all results presented here the value was chosen as $r_c = 2$. Since the radius values are used to calculate the mapping from the DVS to the IF2DWTA their "unit" is in pixel of the DVS. The result of the weight function Eqn. 3.1 is used to set the probability

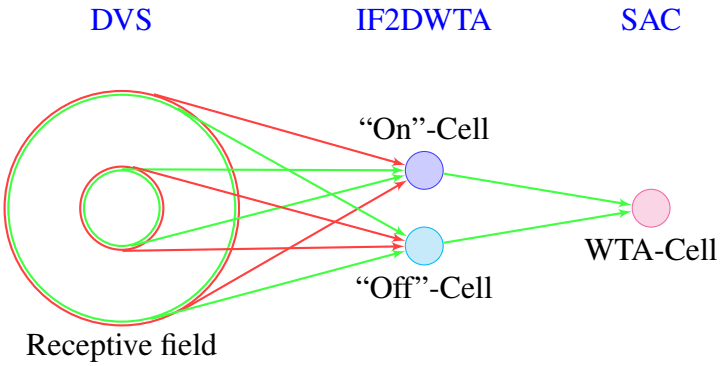
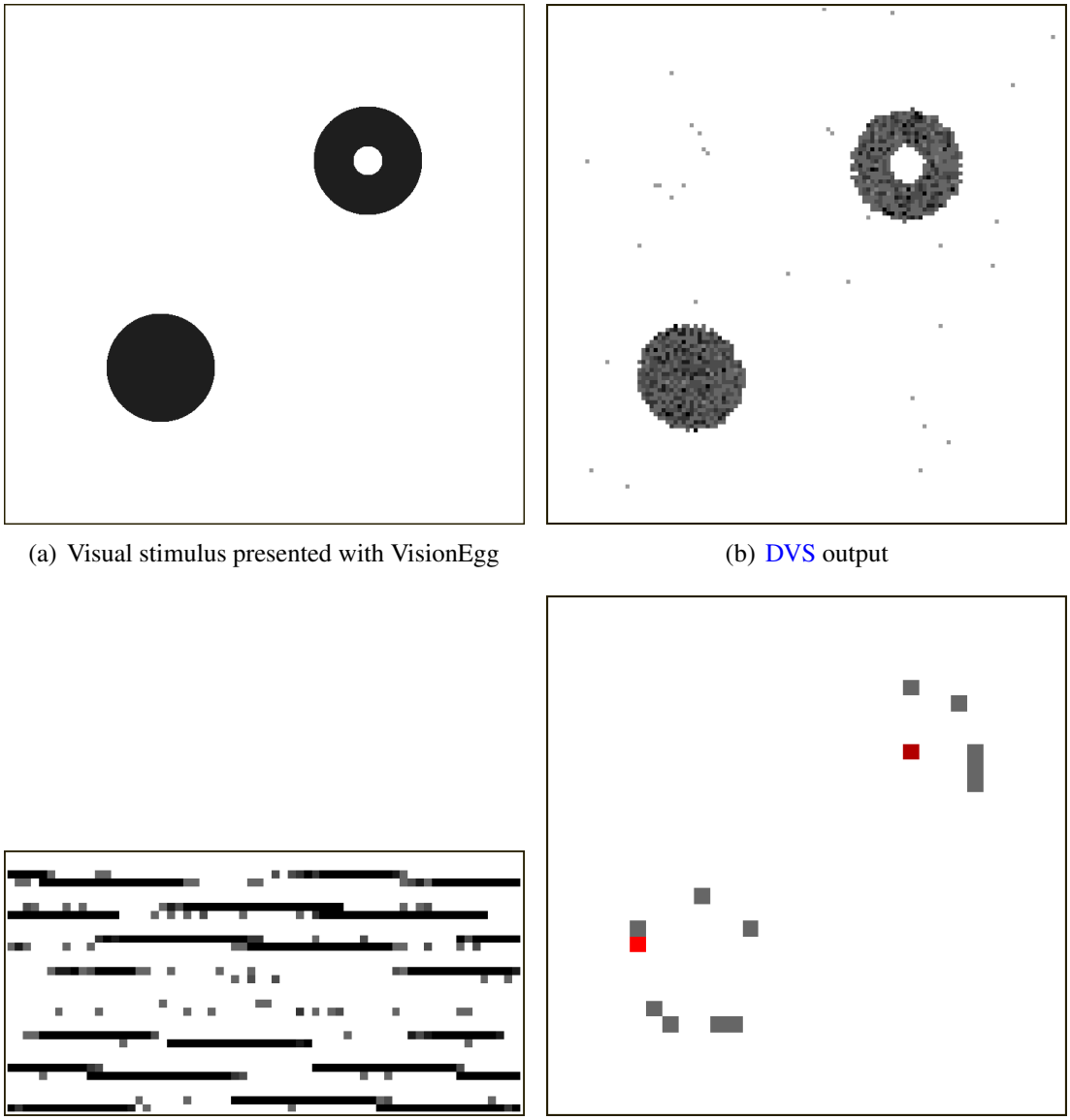


Figure 4.12: Schematic of the neuromorphic attention system's mapping. A pair of "on"- and "off"-cells receives input events from their receptive field generated by the DVS. These connections are weighted according to Eqn. 3.1. They carry out a center-surround operation. The resulting events are sent with equal weights to the same cell on the SAC. Green arrows represent excitatory connections whereas red arrows stand for inhibitory connections. The connections between the different neuromorphic devices is established by a mapping device.

value in the mapper. Its maximum – corresponding to a probability of 1.0 – is 127. That is the value I set for k_c. The method of creating a mapping from the DVS via the IF2DWTA to the SAC works as follows:

- Pick a cell on the SAC, $\langle x_{sac}, y_{sac} \rangle$, within the limits given due to the lack of neurons on the IF2DWTA as explained above.
- Pick the next two not yet used cells on the IF2DWTA. One is used as the "on"-cell, the other as the "off"-cell. Create a mapping from both of these cells to the cell on the SAC.
- The DVS provides 128×128 pixel whereas the SAC has 32×32. Therefore, one can multiply the SAC's coordinates by 4 to calculate the center location on the DVS: $\langle x_{dvs}, y_{dvs} \rangle = 4 \cdot \langle x_{sac}, y_{sac} \rangle$.
- Increase the distance from this center location in both x- and y-direction until the absolute value of the weight function Eqn. 3.1 (with $r = \sqrt{x^2 + y^2}$) is smaller than 1. For each of these pixels calculate both weights $W_{on}(r)$ and $W_{off}(r)$. For both cell types create mappings according to this rule: If $W(r)$ is bigger than zero create an entry in the mapping table from the DVS pixel to the excitatory synapse of the picked "on"- or "off"-cell on the IF2DWTA otherwise create a mapping to its inhibitory synapse.

These rules create the mappings for the neuromorphic attention system itself. I added several monitoring mappings for documentation purposes. As shown in Fig. 4.11 I recorded the output of the DVS, the input events to the IF2DWTA (both represented by the first arrow in Fig. 4.11), and the output of the center surround operation. In Fig. 4.13 a snap-shot of all of these event streams of an experiment is shown.



(a) Visual stimulus presented with VisionEgg

(b) DVS output

(c) IF2DWTA's input and output

(d) Input and output of the SAC

Figure 4.13: Snap-shot of the stimulus and the event streams within the neuromorphic attention system. (a) shows an example stimulus created by the VisionEgg script with a preferred stimulus in the right upper corner and a non-preferred stimulus in the lower left corner. (b) shows the stimulus' recording by the DVS. White pixel represent an increase in contrast, black a decrease. The weighted input (black) and its output (red) to the IF2DWTA is shown in (c). Since the input is stronger than the output no output can be seen. Nevertheless there is output because (d) shows input (black) and output (red) events of the SAC.

4.3.2 Experiments

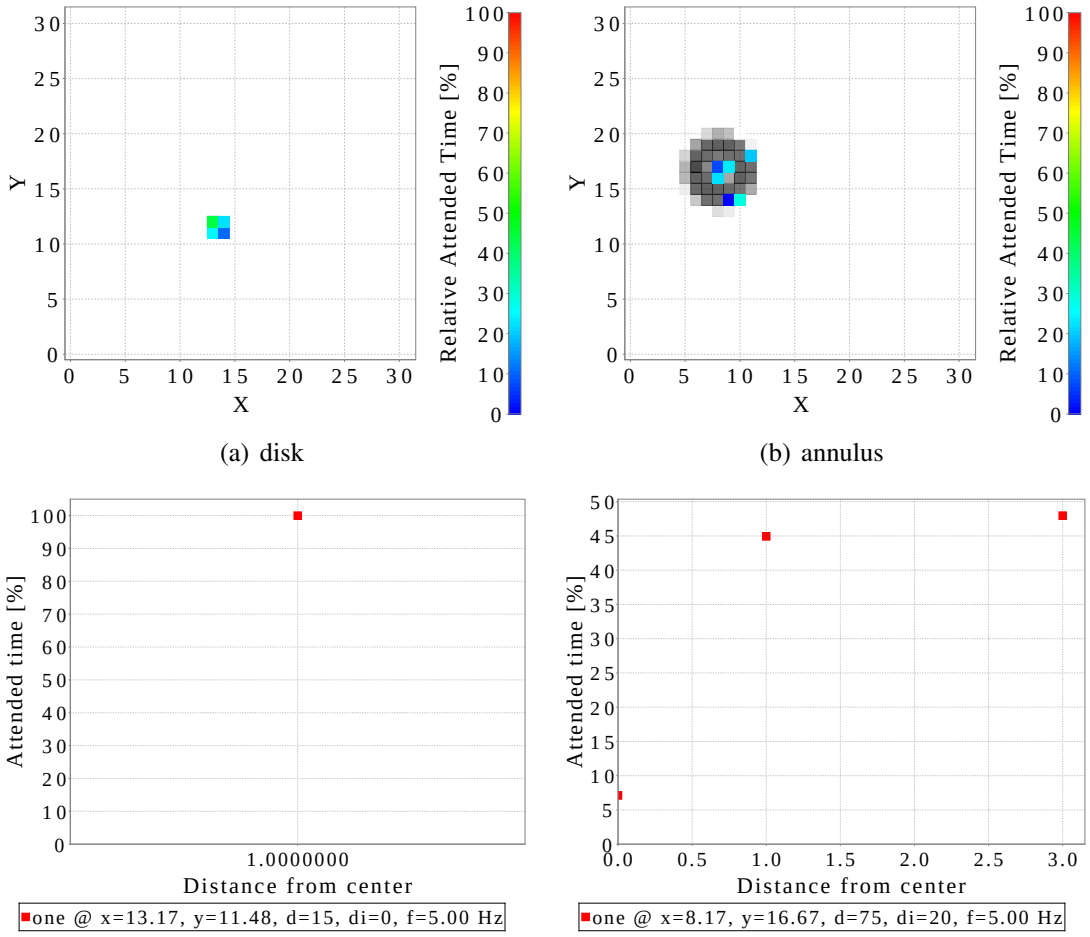
Before each experiment both chips were reset. Then the stimulus was presented on the monitor. All described event streams were recorded for 5 s. Similar to the experiments carried out to investigate the center-surround operation the beginning and the end of the recorded event streams was discarded so that the analysis was computed for only 3 s of data. This was done to avoid on- or off-set artifacts in the data.

The neuromorphic attention system's output is the output of the SAC: The chip creates events with the address of the location with the highest input. Because the mapping from the DVS to the SAC is "retinotopic", by knowing the output address one can infer the most salient region on the DVS' input. Due to its IOR mechanism the SAC inhibits the location that it chose recently to be able to choose the location with next highest input. This inhibition is limited in time so that, if nothing changes on the input side, the SAC will re-choose a location. If a stimulus at location A is stronger than one at location B the SAC will fire for a longer period of time at location A than at location B (compare Bartolozzi [2007, Sec. 5.3.3]). I call the time where one location is attended, "attended time". By putting this time in relation to the experiment's duration one can calculate a "relative attended time" in %.

To compare the outcome to different stimuli at different location I calculate the difference from the center of the stimulus to all attended location reported by the SAC rounded to full SAC pixel. Rounding to a full pixel value is equivalent to binning different values. If there is more than one stimulus the calculation is repeated for all presented stimuli with respect to their center. The stimuli centers are estimated by a K-means clustering algorithm [Bishop, 2006, Sec. 9.1]. When presenting an ideal attention system with one stimulus with highest saliency in the center the outcome would be 100% relative attended time at zero distance. If two such stimuli are presented the values would be 50% at zero distance and 50% relative attended time at the distance of the other stimulus. With this metric it is possible to compare different stimuli at different locations.

Since one stage of the neuromorphic attention system is implemented by the center-surround operation presented in Sec. 3.3 the two preferred stimuli are a small disk, activating the "on"-cells, and an annulus with a small blank inner disk and a bigger active surrounding. The second stimulus attracts attention via the "off"-cells. As shown in Sec. 4.1.2, the term "small" refers to 20 pixel of diameter on the monitor whereas "big" corresponds to 75 pixel of diameter. Fig. 4.14 shows an example output of the attention system when presented these preferred stimuli by themselves. Since the small disk only activates one area on the SAC it is obvious that the system attends to this location for the whole experiment's time. In the case of the small annulus the center-surround operation not only is active for the stimulus' center but also feeds input to the SAC at the edge between the outer radius and the background. Nevertheless, the system attends most of the time to the center of the annulus.

More challenging for the neuromorphic attention system are experiments with more than one stimuli presented at the same time. Fig. 4.15 shows the attention system's outputs for an annulus (preferred stimulus) and a big disk (non-preferred stimulus) of a single experiment run. In the here presented examples the stimuli were always presented in diagonally opposite quadrants of the visual space. In all experiments the attention system attended for some time to the center of the preferred stimulus. As can be seen by this example the lower half of the



(a) disk (b) annulus

(c) disk (d) annulus

Figure 4.14: Preferred stimuli presented to the neuromorphic attention system. X- and Y- coordinates show the ones of the SAC. In gray, only visible in case of (b), the input recorded by the DVS but converted to SAC coordinates is shown. Color coded and superimposed to the input is the relative attended time. The plots (c) and (d) show relative attended time in relation to the distance from the stimuli's centers. In case of the disk the attention system attends 100% of the time at a distance of 1 pixel of the disk's center. In case of the annulus ((d)) the attention system attends to either the center region (about 52%) and to parts of the outer edge (about 48% at a distance of 3 pixel).

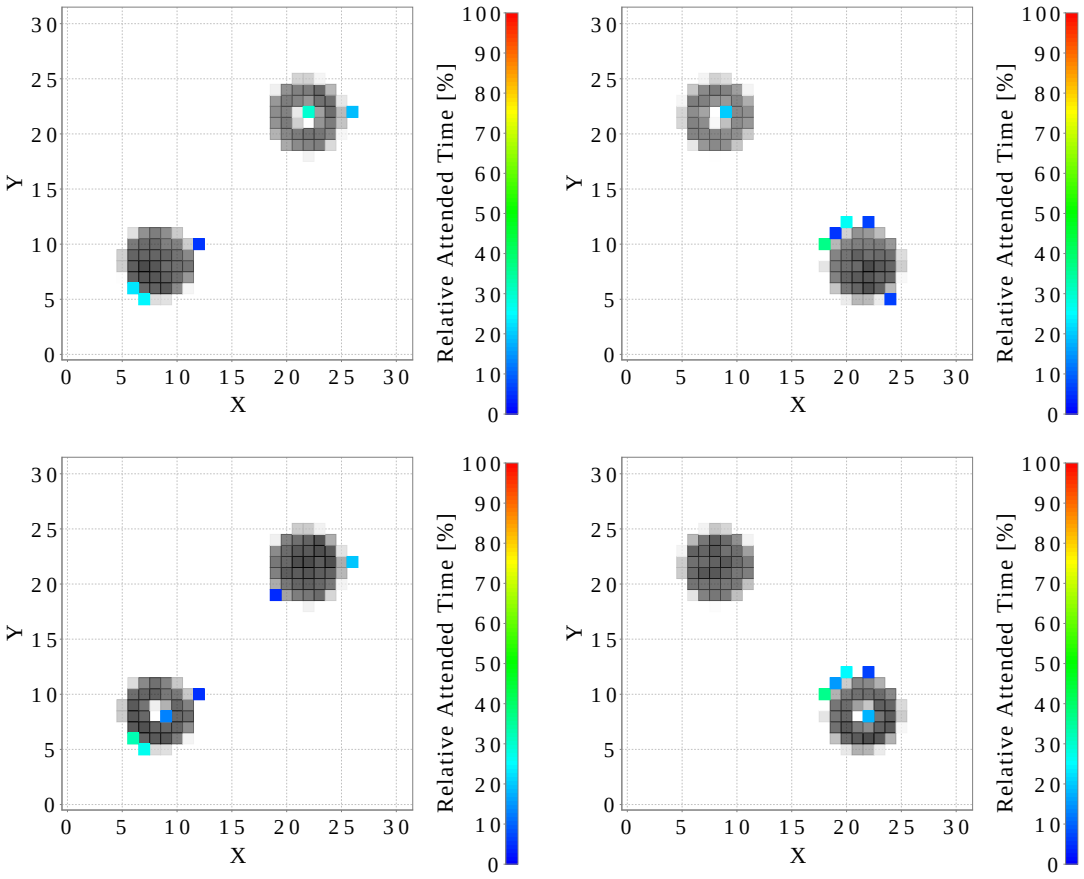


Figure 4.15: Preferred vs non-preferred stimulus. Colored pixel are the output of the neuromorphic attention system stimulated by an annulus (stimulus "two", preferred stimulus) and a big disk (stimulus "one", non-preferred stimulus) always presented in diagonally opposite quadrants of the visual field. The stimuli's intensity is shown in the gray values. Color coded is the pixel's relative attended time.

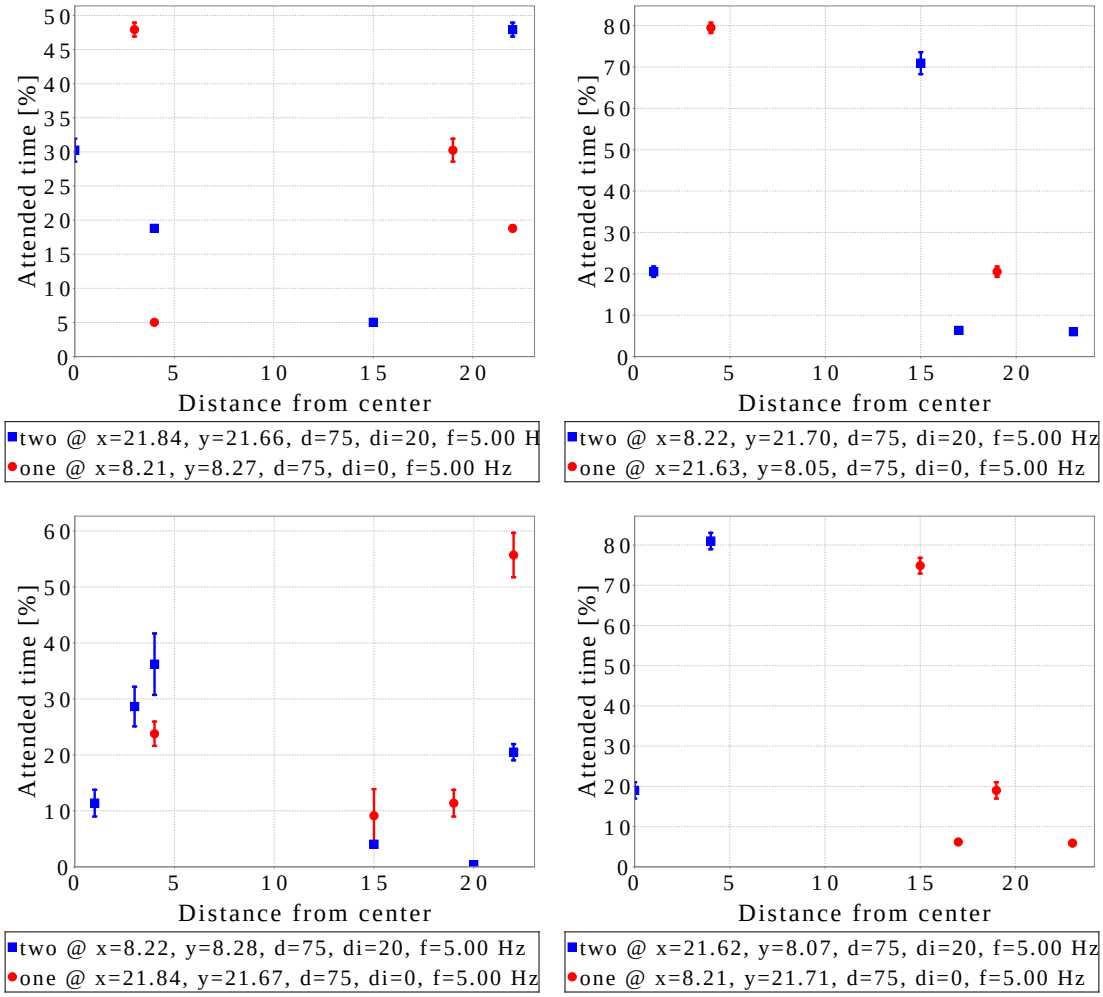


Figure 4.16: Preferred vs non-preferred stimuli: distance from attended pixel to stimulus' center. X-axis represents the distance measured in SAC's pixel from the attended SAC pixel to the stimulus center. The values on the Y-axis show the relative attended time(s) for a given distance. In all four stimulus constellations the attention system chooses the center of the preferred stimulus ("two" – blue markers) for some time. The center of the non-preferred stimulus is never chosen ("one" – red markers). I classify the distances 0 and 1 pixel as center. The edge of the here presented stimulus is about 3 to 4 pixel away from the stimulus' center. The statistics result from showing the same stimuli in 5 separated experiment runs.

system is more sensitive to inputs than the upper half. The system especially attends for a large amount of time to pixels in the lower right area. This is most likely due to mismatch in the neuromorphic chips used in the system. Nevertheless, I can show by these examples that the system chooses only locations with high saliency: Fig. 4.16 shows that the system attends to the inner of the annulus between 12 and 30% of the time. During the rest of the time the system attends to the stimuli's edges. There is no case where the system chooses the center region of the big disk even though the input rate to those pixel is quite high as can be inferred from the high gray values in Fig. 4.15.

4.3.3 The center-surround operation is essential for the system

The neuromorphic attention system presented in the last section is able to detect salient regions in the input space reliably. This was tested by a series of experiments presenting preferred and non-preferred stimuli to the system.

In the experiments presented the stimuli did not move but flicker. This contradicts the essential idea behind the DVS. The visual sensor is designed such that it generates only events when it detects a change in its visual field. If the DVS is fixated this implies that it generates events mainly at the edges of a presented moving object. Because edges are considered as salient regions feeding these events directly to the SAC to let it choose the most salient ones would already fulfill the requirements for an attention system. This is the approach presented in Bartolozzi [2007]. Nevertheless, I argue that the system I presented here is superior to one that omits the center-surround operations: Similar to our eye if an object does not move by itself the DVS is not able to detect it since there are no changes in contrast. In biology this issue is solved by constantly moving the eyes by a small amplitude. This motions are called micro-saccades [Martinez-Conde et al., 2004, 2006]. It is equivalent to moving an object in front of the DVS by a small amplitude. Doing so for a textured object will generate similar event streams as a blinking object hence it is necessary to extract edges from this stream. This task can only be accomplished if center-surround operations are available.

4.4 Conclusion

Throughout this chapter I present both preparatory experiments and experiments conducted with the multi-chip neuromorphic attention system and their results. In the first part I focus on the center-surround operations implemented with the help of the IF2DWTA. First, its synapses' behavior is investigated with synthetic spike trains generated on the workstation. This is followed by experiments where the events are generated by the DVS when it is recording stimuli presented on a monitor. The second part discusses the SAC and its determination of salient regions in detail. The section is split up in covert and overt experiments. All these experiments laid the foundation to build a neuromorphic attention system. This is described in detail before experiments and their results showing its capabilities are presented.

5 Discussion & Conclusions

The final chapter summarizes the topics covered by this thesis. It puts the system developed throughout this thesis in context of artificial vision. The goals that this this thesis achieves are described. Next the chapter presents the roots of the neuromorphic selective attention system and its competitors. Even though the system in its current state is a proof-of-concept I speculate about its future impact and applications. Finally, the chapter gives an outlook of further possible improvements.

5.1 The system's context: Vision

Visual perception, or vision, is the ability of an organism to extract from electro-magnetic waves, i.e. light, information about properties like e.g. location, shape, color, size, and texture of an object or its environment. There are two separate functionalities necessary to enable vision: First, the emitted or more likely the reflected light from an object has to be transformed into a processable representation. Second, this raw data has to be processed such that the relevant information is extracted. The relevance of information is usually task dependent: If an organism is looking for food, relevant information might be the red little objects in a bigger green one. Some other information is important irrelevant of the task the organism is performing: An example are quick motions that can be signs for the presence of a predator. In biology several different visual systems evolved, from the rather primitive pinholes found in mollusks [Land and Fernald, 1992] in correspondence with a primitive nervous system up to the human visual system.

The human visual system still outperforms any artificial visual systems: For example the human visual system is able to detect faces very quickly and still very accurately (almost 95% accuracy) [Crouzet et al., 2010] compared to today's computational vision systems that achieve accuracy values of 50-70% [Zhang and Zhang, 2010]. If the system that contains the artificial visual system is a mobile one, e.g. a mobile robot, not only the ability to process visual information per se has to be considered but also the necessary power. The problem is not the generation of a processable representation of the scene. With todays technology low power image sensors that are built for example into mobile phones are able to provide video streams of 1920×1080 pixel resolution at 30 Hz [Barczok et al., 2012]. These are cheap and are often used as sensors for mobile robots.

The problem is the second step of vision, the extraction of the relevant information from the huge amount of visual data provided by its sensors in useful time, i.e. real-time. The latter requirement is necessary so that the system is able to react immediately on changes in its environment. Since we are considering mobile systems this extraction has to be performed under the constraint of low power dissipation.

In this thesis I presented one possibility to implement an important sub-system towards a relevant information extraction system: the selective attention system. In Kosslyn et al. [1990] a model of high-level vision is described. In contrast to low-level vision that is purely input driven, high-level vision uses previously stored information to carry out tasks like object recognition and navigation. Therefore, the authors identify several "processing subsystems" and define their computational tasks. One important sub-system is the attentional window: The part of the visual buffer that lays within the attentional window is processed further by other sub-systems. Since the computational resources are limited it is beneficial to process mainly relevant information. As already mentioned the information that is relevant is on the one hand task dependent on the other hand influenced by what the visual sensor records. In the literature these two influencing factors are called *top-down* for the task dependent and *bottom-up* for the input driven influence, respectively [Neisser, 1967]. In this thesis I only considered bottom-up based selective attention – nevertheless as shown in Sec. 4.2.1 the incorporation of a top-down signal in a future systems making use of the system presented in this thesis is possible as well.

To my knowledge the system presented throughout this thesis is the first visual selective attention system using neuromorphic chips for the detection of salient regions. It implements the most important computational stages of Itti et al. [1998]'s model: This is the detection of spatial discontinuities with the help of the center-surround operation and the selection of the most salient regions by a WTA network in combination with IOR. This is achieved by two neuromorphic chips: the IF2DWTA and the SAC.

5.2 The presented system achieves different goals

The presented work is an important step in different directions:

- The neuromorphic selective attention system is an example to show the computational capabilities that arise from the combination of different neuromorphic principles: First, the attention model is inspired by biological methods [Koch and Ullman, 1985]. Second, the used devices emulate biological neurons and synapses. Furthermore, the computation is carried out in the emulated synapses and neurons whereas the communication of the system is based on spikes. This use of both, analog computation together with digital communication, was inspired by the brain. All these methods are combined within the selective attention system that is able to identify salient regions in visual scenes in real-time.

- The system presented is the first implementation of an attention system with neuromorphic chips to my knowledge. It is the successor of a series of systems based on work by Indiveri [2000b]. These systems focus only on the last stage of Itti et al. [1998]'s model: the creation of the scan path. They can only be classified as attention systems if the input to the chips is a saliency map. In case of the presented system the saliency map is computed for one feature by a neuromorphic chip. Even though the ability to identify salient regions with different size is still missing, the center-surround operation imple-

mented on the IF2DWTA detect spatial discontinuities. This ability sets the presented solution apart from other neuromorphic implementations.

- The selective attention system presented in my thesis is an example of a neuromorphic engineering solution to a practically relevant problem: Its purpose is to guide an artificial vision system to salient regions for further investigation. Several research groups in the world develop hardware emulating synapses and neurons [Furber and Temple, 2007, Painkras et al., 2013, Choudhary et al., 2012, Brüderle et al., 2011]. Systems with up to one million neurons are proposed [Silver et al., 2007] that will provide immense computational power. Despite this computational power the problem arises to make practical use of this hardware. The neuromorphic attention system presented in this thesis is one example where neuromorphic hardware is used to solve a practical problem in an engineering fashion.

Nevertheless the system presented throughout this thesis accomplishes the mentioned goals it is still in its current state a proof-of-concept. To be operable it needs two workstations: One controls the parameters of the neuromorphic chips and is used to monitor the events transmitted throughout the system. The second one provides its main memory for the look-up table of the probabilistic mapping device. Furthermore the system's neuromorphic chips are mounted on different PCBs that are interconnected with cables. All this equipment is dedicated to enable flexibility. This flexibility allowed the engineering of the system. The downside of this flexibility is the high power dissipation caused by the workstations and the bulkiness of the whole system. In a future version all these elements can be integrated into a compact, low-power system able to detect salient regions in a visual scene recorded by a DVS. This can be accomplished by creating one PCB carrying the two neuromorphic devices, a small microcontroller for the control of the parameters and a FPGA for the mapping. Even a more integrated version of the system can be engineered by building a single VLSI device consisting all necessary elements. By this proposed integration the system could provide its functionality e.g. for mobile robotic systems.

Itti et al. [1998]'s model is able to detect salient regions of different scales. This multiscale feature extraction is not implemented by the system presented. Similar to the underlying model it could be implemented by using several IF2DWTAs with different sizes of receptive fields to compute the center-surround operations. The output of the neuromorphic chips would then be fused and sent to the SAC to create the scan path. The fusion of different AER streams works well as described in Sec. 4.2.1.

5.3 The system in its historic context

Today, computers help us in our daily life in very different forms: From smart phones, notebooks, desktop computers, and servers up to high-performance computers for scientific or military purposes. The computational heart of all these different forms of computers are based on similar technology: On the hardware level it uses transistors as switches. In digital systems they are either on or off. The computational logic circuits are subdivided into different building blocks. To forward the results from one building block to the next they have to be

synchronized. This synchronization is achieved by a clock signal distributed over the whole chip with high accuracy.

On the architectural level today's digital technology still follows the von-Neumann-architecture [von Neumann, 1993]: A Central Processing Unit (CPU) reads and writes data and execution instructions from a separated memory. This memory is connected to the CPU with a bus that transmits data between the processor and the memory.

This technology has several advantages: Because the transistors are used as switches the fabrication of these devices is quite robust to fabrication mismatch. The design of such devices is very well supported by CAD systems and therefore relatively simple to achieve. Due to the relatively simple architecture the complexity to develop software for these systems is low. Another advantage of the architecture is the ability to extend it quite easily: Since the communication is done by a bus system, external devices can be integrated into the system by connecting to this bus. Also the system's computational power was increased impressively over the last six decades [Moore, 1965, Mack, 2011].

But its main advantages are also the source for its main disadvantages: To provide a stable clock signal that synchronizes the whole chip quite some effort is required. Both the design of such a clocking system is difficult as well as the operation is costly. In today's digital chips the generation and distribution of this signal dissipates a major if not the biggest amount of energy used by the device [Gronowski et al., 1998]. Because the computational power increased much quicker than the bandwidth of the bus that connects the CPU to the memory the bus system turned into a bottleneck of this technology [Backus, 1978]. Improvements like caches, small on-chip memories that buffer data and instructions, cannot solve the inherent architectural downside.

To overcome this technological and architectural limitations, a totally new computational approach is necessary. One possible solution is neuromorphic engineering. Taking the most powerful computational device known, the brain, as example neuromorphic engineering offers a possibility to built powerful computational devices beyond today's von-Neumann's architecture based computers. The school of neuromorphic engineering was established in the late eighties by Mead, Mahowald, and others: Because the brain exceeds today's computers in many day-to-day tasks like vision or audition using neurons and synapses, as the computational primitives, to carry out computational tasks seems a promising approach. Instead of using digital computers to simulate neurons and synapses the researches proposed to emulate neural substrate by analog circuits. The idea to use VLSI technology to emulate neurons and synapses was based on the observation that the physical properties of transistors operated in the sub-threshold or weak inversion regime are similar to the ones of neuron channels [Mead, 1989]. A summary about neuromorphic engineering's history and goals can be found in Indiveri and Horiuchi [2011].

From the hardware point of view neuromorphic engineering offers very low power dissipation. This has several reasons: First, the computational primitives, neurons and synapses, are not numerically simulated but emulated by circuits. Second, the transistors of these circuits are operated in sub-threshold or weak inversion. Because currents are very small in this operational regime the power dissipation is very small compare to digital circuits. From an architectural point of view the computation is carried out in a highly parallel fashion in the neurons. There is no clock signal to synchronize the computation amongst neurons. They

do their calculation whenever an event arrives at their synapses independent of its surrounding. Furthermore, the data is not stored in a memory far from the computational unit(s) but just where the calculation is carried out: In the weights of the synapses and the membrane potential of the neurons.

Despite these advantages neuromorphic engineering has another side of the coin: The amount of current that flows through a transistor encodes a signal, and not only if there is current or not like in digital systems. If two values should be compared to each other it is important that the same value generates the same current nevertheless which circuit it generates. Due to fabrication mismatch this basic requirement is not met. Therefore it is very difficult to build neuromorphic systems to carry out exact calculations. This effect can also be observed if emulated neurons with the same set of parameters are driven with the same input current: Their output frequency will differ by a considerable amount [Indiveri and Chicca, 2011]. There are several possible ways to deal with this mismatch: One possibility is to reduce it by using bigger transistor sizes and other design approaches. This comes at the cost of the need of more silicon area. Another, approach is to compensate for the fabrication mismatch. Neftci and Indiveri [2010] uses probability based connections from the input to the output layer to correct for mismatch. Another approach is to use a probabilistic mapping device as described in this thesis in Sec. 2.1.3 or in more detail in Fasnacht and Indiveri [2011]. Both of these compensations are not satisfactory due to the non-linear properties of neurons. They can only compensate the effects of mismatch for one operating point. Instead of compensating the mismatch Sheik et al. [2012] makes use of it by choosing carefully certain neurons with a distinct firing rate to implement axonal delays.

If considering the biological example neurons in our brains also differ in their output behavior when stimulated with the same input. Nevertheless our brain is able to cope with this mismatch very well. This suggests that the algorithms the brain uses are either robust enough to overcome this mismatch or the brain constantly adapts the connection weights to improve its abilities. As shown in this thesis applying models onto neuromorphic hardware is a challenging task also because of fabrication mismatch. The question arises if evolution tuned the neurons in the brain such that they can execute these models? This assumption seems unlikely because if the brain is damaged most function can be recovered. In these cases other brain areas overtake the functionalities of the damaged ones. [Robertson and Murre, 1999]. This suggest that instead of a tuning done once and forever the brain constantly adapts the weights of the connections between the neurons to accomplish its tasks. This tuning is usually called learning. Hence, in a future version of the proposed selective attention system an inclusion of a learning mechanism in the connecting weights might improve the robustness of the system.

Further difficulties with neuromorphic engineered systems raise when algorithms should be established on neural networks. Because there is no central memory the programming is quite difficult. Furthermore most algorithms were developed for conventional computers. Often they cannot be applied to the new way of computation. Therefore new methods have to be developed. Also software development tools are not available yet. A possible mean to overcome these shortcomings is proposed by Eliasmith and Anderson [1999, 2004]: they developed the Neural Engineering Framework (NEF), “an approach to built large-scale biologically plausible models.” [Eliasmith, 2007] With this method the authors are able to generate neural network models that can execute advanced tasks like character recognition and reproduction [Eliasmith

et al., 2012]. Neftci et al. [2011] propose an alternative method to map the parameters of a neural model onto neuromorphic hardware. Despite these improvements programming a conventional computer is still easier than deploying neural models on neuromorphic hardware.

Given the advantages and disadvantages of both the conventional computational approach following the von-Neumann-architecture and the neuromorphic engineering approach I assess the possible application for engineering problems depending on the time horizon: Because of the difficult deployment of algorithms onto neuromorphic hardware and the necessary handling of fabrication mismatch I advocate a mixed approach for a short to medium time horizon: Taking advantage of the low power, event driven possibilities neuromorphic sensors offer I would use them as input data source for an engineered system. For the computational part I still think it is beneficial to use today's low power yet powerful digital VLSI technology. By taking advantage of both easier programming and debugging one can quicker and cheaper provide solutions for engineering problems. An example is the pencil balancer developed by Conradt et al. [2009]. Even though this system cannot be seen as a commercially successful product it still shows the power of the mixed approach: two DVS sensors provide visual input information at a high rate with low bandwidth so that a cheap, low power microcontroller can do its calculation to balance a pencil with actuators. If the construction of a brain like computational device within a longterm time horizon is the goal I opt clearly for the neuromorphic engineering approach. But before such a device can be realized for engineering problems many open issues still have to be solved. Nevertheless taking the arguments from Douglas et al. [1995] building a computational device that can compete with the brain extreme low power consumption is required which cannot be accomplished by conventional digital VLSI technology but with neuromorphic engineering. For such a device also the von-Neumann-bottleneck problem can only be solved by distributing the necessary memory over the whole device.

The system presented in this thesis is an approach beyond a pure sensor: The system consist of a neuromorphic visual input sensor and the first computational steps of a vision system as described in Kosslyn et al. [1990]. In fact it is an important step towards a full implementation of Itti et al. [1998]'s selective attention model by using only neuromorphic chips. The system is a successor of a series of systems all using neuromorphic devices: Indiveri [2000b] used the first three stages of a software implementation of this model to generate a 2-D-saliency map. This map was used as an input for his neuromorphic 1-D-WTA network. The chip detected the location where the input had its maximum. A IOR mechanism allowed the de-selection of the current winner and the selection of other locations with high input. Even though this chip can only process one dimensional input vectors it already offers most of the functionalities that the SAC provides. Nevertheless the input to the chip needs to be a saliency map to operate as an attention system. An extension of that work was presented in Indiveri [2000a]: The chip offered now 64 input synapses arranged in two dimensions (8×8). Another improvement was developed by Bartolozzi [2007]: She extended the functionality of the input synapses with short term depression and implemented an adaption circuit for the output neurons. At the same time her Selective Attention Chip (SAC) grew in size and provides 32×32 synapses for input signals [Bartolozzi and Indiveri, 2009]. Albeit these improvements the SAC still needs a saliency map as input to be used in the context of an attention system. This is the main improvement of the system presented: It extracts its saliency map from the

input stream provided by the DVS. This is achieved by center-surround operations executed by the neuromorphic IF2DWTA chip.

In Itti et al. [1998] salient regions are detected on the basis of three features: light intensity, color, and orientation. The input sensor of the system presented is the DVS that detects temporal contrast. Hence this is the only feature that is used to decide if a region in the visual space is salient. In Wolfe and Horowitz [2004] the authors list features that were identified to play a role in the detection of salient regions. They point out that for several features including motion there is no doubt that they are very important cues to separate salient from non-salient regions. Objects that move within the visual field of the DVS change contrast and are therefore detected. Hence a quick moving object or a rigid object in a flowing environment are both salient regions that are detected by the system presented. Nevertheless in a future version the inclusion of other features would improve the ability to detect salient regions. Source for such features could be a DVS that is able to detect color [Berner, 2011] or the inclusion of another neuromorphic chip that is able to extract orientation information from the DVS output [Chicca et al., 2007].

5.4 The system compare to state-of-the-art

In the next section the system presented in this thesis should be compared with other selective attention systems. This comparison is on a qualitative basis and not quantified with hard numbers. This is due to the fact that the system presented in the thesis is only a proof-of-concept. For its operation it uses two workstation: One to control the parameters of the neuromorphic chips and to monitor the events transmitted between the different devices and another one to map the events between the neuromorphic devices. In a future solution that should be applied e.g. for mobile robotic applications these workstations should be replaced by microcontrollers or FPGA logic devices. Another reason is given by the different working philosophy of the neuromorphic devices. The DVS e.g. emits events representing a detected change in contrast of a pixel whenever this change happens. In contrast common vision sensors are frame based. A comparison of these two approaches is not possible.

The model of Itti et al. [1998] is used as the basis for software libraries, e.g. for C++ [Itti, 2004] or Matlab [Walther and Koch, 2006]. These libraries were first designed to process static images but were also used in systems that process video data later. One of these implementations uses a Beowulf cluster to detect salient regions in a video stream at 30 frames/s at a resolution of 320×240 pixel [Itti, 2002]. A Beowulf cluster is a cluster of common personal computers, usually of the same type, interconnected with a common local area network to execute tasks in parallel. The author uses a system with 10 nodes and 16 CPUs. Other than the system presented in this thesis the Beowulf based system cannot be classified as a mobile system due to its power consumption and its bulkiness. Its advantage in comparison to my neuromorphic system is the flexibility due to the use of common computer technology and the complete implementation of the model. In the same publication the author presents another system that applies the software developed for the Beowulf cluster on a mobile robotic platform: For its computations it uses four CPUs on two boards interconnected with Gigabit Ethernet. Power is provided by two lithium battery packs. The author does not comment on

run times or power consumption hence a comparison with the system presented in this thesis is not possible. With this mini-Beowulf cluster the system is able to process a 160×120 pixel video stream at 30 frames/s. The resolution that this system processes is similar to the resolution provided by the DVS (128×128).

In Frintrop et al. [2007] the authors present an implementation of an extended version of Itti et al. [1998]'s model [Frintrop, 2006] running on a single workstation. By optimizing their code and the method to compute the important center-surround operation they were able to process images of half-VGA resolution at about 15 Hz [May et al., 2007]. To compute salient regions within the visual input the system still needs the computational power of a workstation. Hence the necessary power requirements are still way higher than what could be achieved by an integrated neuromorphic system. May et al. [2007] report a system based on the algorithm by Frintrop [2006] that uses the computational resources of a GPU. By parallelization of the tasks to identify salient regions they could speed up the selection process such that they can process VGA images at a speed of 30 Hz. Nevertheless these improvements they still need a workstation.

A similar system was presented by Xu et al. [2009]: They are using Itti et al. [1998]'s original model and are using the computational power of up to four high performance graphic cards. The authors are able to process VGA images at speeds up to 313 Hz at the costs of up to 450 W for each GPU. Nevertheless with the help of high power polymer Li-Ion battery packs they achieve to put this computational power on a mobile robot. Due to their immense computational power and the possibility to parallelize the tasks necessary to compute salient regions GPU based systems are able to fully implement the model by Itti et al. [1998]. Their main drawback are the high power requirements.

The systems discussed so far were all based on general purpose workstations. In Ouerhani et al. [2002] and Ouerhani and Hügli [2003] the authors present a selective attention system implementing Itti et al. [1998]'s model partially based on a compact, low-power Single Instruction Multiple Data (SIMD) system, called ProtoEye [Ruedi et al., 1996]. They implement all stages of Itti et al. [1998]'s model but only for one feature: image intensity. This implementation is executed on 64×64 processing units each comprising of a 4 bit Arithmetic Logic Unit (ALU), six registers and an analog diffusion network. It is able to process about 14 frames/s. Compared to the system presented in this thesis this system shares similar properties: It is low power, using both digital and analog circuits and implements the attention model partially. In fact the implementation of the system presented in this thesis is not as complete as Ouerhani et al's system since it lacks possibility to detect salient regions of different sizes.

The system presented in this thesis is a proof-of-concept. Therefore I compared an integrated future version with several state-of-the-art implementations. The ones using the computational power of one or more workstations and/or their GPUs implement Itti et al. [1998]'s model completely. Hence their ability to detect salient regions in the visual input is much more advanced as the abilities of the system presented in this thesis. This comes at the cost of high power dissipation and bulkiness. Ouerhani and colleagues implemented parts of the model on a specialized hardware: They are using a SIMD processor in combination with an analog diffusion network. With this system they are able to built a compact, low power system able to predict salient regions based on the implementation of all three stages for one feature.

5.5 Possible application of the system

The system presented throughout this thesis is a proof-of-concept. In its current status its only purpose is to demonstrate the possibility to create a visual selective attention using neuromorphic devices. The system is an example of a neuromorphic system that carries out not only some model's emulation but can be used for a practically relevant task: to identify salient regions in a visual scene. A future version of the selective attention system that integrates all three neuromorphic devices and the necessary mapping could be beneficial for several systems. Surveillance systems could better identify regions that should be watched. Both conditions are covered: the attention system is able to identify moving persons or objects in areas where no one should stay and could also detect stopping persons within a moving crowd, e.g. in a passage way. In mobile robotic applications it is important that each sub-system dissipates as little power as possible. Therefore a system that selects important regions from the visual input space is very beneficial. The visual computation system can then focus on the most important parts of a visual scene and discard the rest. Another application for a visual selective attention system could be in an online video encoder. Before encoding the signal is fed to the attention system that defines regions that are salient and therefore more likely to be watched by the user. These regions could then be encoded with higher accuracy than the rest of the image. With this system both bandwidth and computational power could be saved.

5.6 Outlook: Next possible steps

The system presented can be extended in different directions. Four possible extension are:

- In the first stage of Itti et al. [1998]'s model the image is filtered linearly using different features – color, intensity, and orientation – to create feature maps. These maps are subsampled with Gaussian filters to obtain maps at different scale levels. This is done to identify salient regions in the input image of different sizes. For the center-surround operation that detects spatial discontinuities in the input I used only one set of radii. Therefore my system can only detect salient regions at a certain size. This is due to the limiting factor of available inhibitory synapses on the Integrate & Fire 2–Dimensional WTA (IF2DWTA) chip that carries out this operation. In an extended version one could use more chips of that kind to be able to implement an attention system which is able to detect salient regions in the input space at different sizes.

- The presented system is able to detect salient regions based on the input provided by the DVS. The neuromorphic system's processing stages carry out their computation on the basis of spike events. These events are not limited to any type of feature. Therefore the system can easily be extended to incorporate other features. For example by using the cDVS proposed by Berner and Delbruck [2010] would allow to identify salient regions also by making use of the feature "color". The necessary requirement is that for each feature added to the system more computational units are available to execute the center-surround operations. Using more features would also require the investiga-

tion of the combination of different feature maps. This is an ongoing subject of current research [Itti and Koch, 2001b].

- Even though the main focus of this thesis is on *visual* selective attention one could implement an attention system using a totally different modality – like auditory – with the same processing stages. To do so two main requirements have to be met, one conceptional and one technical: First, the input to the center-surround operation has to be provided in a map-like structure. This is due to the fact that the model of Itti et al. [1998] relies on maps. Second, the input has to be provided by an AER stream. To meet this requirement the original source of the map does not necessarily output AER events. One can also implement an inter layer that translates the input map into AER events.

- In the presented system the processing is separated into two chips: the IF2DWTA performs the center-surround operation whereas the SAC creates the scan path. In a future version of the system these two chips could be fused into one single device. This would make the system more power efficient and more compact. The costs of doing so would be: The proposed system would be harder to monitor and it looses flexibility in the mapping from one chip to the next as it is implemented by the AER mapper in the current system.

5.7 Final Summary

Throughout the thesis I described the possibility to implement a selective attention system by making use of neuromorphic hardware.

In the first chapter I review different models of saliency-based visual attention. Due to its popularity I chose the model of Itti et al. [1998] for my implementation. Then I gave an overview of the existing implementation of visual attention methods in technical systems. In this realm exists a big variety of systems: From software implementations running on powerful workstations over implementation with custom designed analog VLSI chips to mixed digital/analog SIMD solutions and neuromorphic implementations.

The second thesis chapter gives an overview over the neuromorphic hardware that is used throughout this thesis. The input sensor is the Dynamic Vision Sensor (DVS). In contrast to conventional vision sensors that are frame based the DVS generates an event at the location in its field of view where it detects changes in contrast. The first processing is then carried out by the IF2DWTA chip. It performs a center-surround operation to detect spatial discontinuities. Finally, the third used neuromorphic device is the SAC. It is used to determine the region with highest saliences. The last two devices' parameters are controlled by an AMDA board. During my thesis I developed a new firmware for this PCB. All these neuromorphic devices communicate amongst each other by the Address Event Representation (AER) bus. This communication is realized by AEX boards. To monitor and stimulate neuromorphic chips from a workstation I developed a client-server-architecture: An AEX server communicates with one of the AEX boards over USB. Programs that carry out experiments are clients that monitor and stimulate the chips via the server. This architecture ensures that several clients can transmit and receive events to and from the neuromorphic multi-chip system at the same time.

In Chap. 3 I discuss three important topics: First, I described Itti et al. [1998]'s model in detail. Second, I rise arguments to support my implementation of an attention system even though it only relies on one feature, namely temporal contrast. To locate spatial discontinuities the model used for this thesis proposes center-surround operations. I explain my implementation of the center-surround operation based on research work accomplished at the cat's retina in the third part. From a biological point of view this approach seems very questionable since no evidence was found how the brain identifies salient regions in the visual space. Since I do not claim that the presented system is an emulation of the biological attention system but rather an engineered system inspired by biological observations the proposed approach seems acceptable.

All my experiments to show the different facets of the system are presented in the next chapter. The chapter is structured such that first experiments conducted with the different building blocks, i.e. the two neuromorphic chips, are introduced before the results of experiments conducted with the full neuromorphic system are described. The experiments with the building block comprises: The IF2DWTA's inhibitory and excitatory synapses' properties are investigated. Then their interaction to implement the center-surround operation are shown. Next, several experiments to show the abilities of the SAC to generate the scan paths are presented.

The system presented is an example for a neuromorphic system that can be used in a practically relevant task: It is able to identify salient regions in its visual input space and can guide another technical system to these spots. The process from theory over the implementation to conducted experiments and their results is described.

Even though the system is based on neuromorphic devices that are proven to carry out computations very efficiently the presented system dissipates a lot of power. This is due to the fact that the system is a proof-of-concept. Engineering such a system requires flexibility. Therefore the system uses serveral PCBs and two workstations. In a future version of the system all these equipment could be integrated onto one PCB or even into one VLSI device.

The ability to carry out center-surround operations makes the presented system superior to approaches omitting these operations. With the help of center-surround operations the system is able to extract edges from objects flickering or moving by small amplitudes. The visual input sensor used for this thesis – the DVS – can only record objects that move or flicker. This is analogous to the eyes that carry out micro-saccades. Edges are considered as salient regions within a visual scene. Therefore the extraction is essential for a system to be classified as selective attention system.

Bibliography

R.A. Abrams and S.E. Christ. Motion onset captures attention. *Psychological Science*, 14: 427–432, 2003.

ATmega128. 8-bit Atmel microcontroller with 128 kbytes in-system programmable flash - Summary. Atmel website, June 2011. URL http://www.atmel.com/devices/atmega128.aspx?tab=documents.

J. Backus. Can programming be liberated from the von neumann style?: a functional style and its algebra of programs. *Communications of the ACM*, 21(8):613–641, 1978.

A. Barczok, H.A. Czerulla, and L. Labs. Das Imperium würde gern zurückschlagen. *c't*, 26: 84–95, 2012.

C. Bartolozzi. *Selective attention in silicon: from the design of an analog VLSI synapse to the implementation of a multi–chip system*. PhD thesis, ETH Zürich, Zürich, Switzerland, May 2007.

C. Bartolozzi and G. Indiveri. Selective attention in multi-chip address-event systems. *Sensors*, 9(7):5076–5098, 2009. ISSN 1424-8220. doi: 10.3390/s90705076. URL http://www.mdpi.com/1424-8220/9/7/5076.

R. Berner. *Building-Blocks for Event-Based Vision Sensors*. PhD thesis, ETH Zürich, Zürich, Switzerland, 2011.

R. Berner and T. Delbruck. Event-based color change pixel in standard cmos. In *International Symposium on Circuits and Systems, (ISCAS), 2010*, pages 349–352. IEEE, June 2010. doi: 10.1109/ISCAS.2010.5537787.

C.M. Bishop. *Pattern recognition and machine learning*. Springer New York, 2006.

V. Brajovic and T. Kanade. Computational sensor for visual tracking with attention. *IEEE Journal of Solid State Circuits*, 33(8):1199–1207, August 1998.

D. Brüderle, M.A. Petrovici, B. Vogginger, M. Ehrlich, T. Pfeil, S. Millner, A. Grübl, K. Wendt, E. Müller, M.-O. Schwartz, D.H. de Oliveira, S. Jeltsch, J. Fieres, M. Schilling, P. Müller, O. Breitwieser, V. Petkov, L. Muller, A.P. Davison, P. Krishnamurthy, J. Kremkow, M. Lundqvist, E. Muller, J. Partzsch, S. Scholze, L. Zühl, C. Mayr, A. Destexhe, M. Diesmann, T.C. Potjans, A. Lansner, R. Schüffny, J. Schemmel, and K. Meier. A comprehensive workflow for general-purpose neural modeling with highly configurable neuromorphic hardware systems. *Biological cybernetics*, 104(4):263–296, 2011.

E. Chicca, A.M. Whatley, P. Lichtsteiner, V. Dante, T. Delbruck, P. Del Giudice, R.J. Douglas, and G. Indiveri. A multi-chip pulse-based neuromorphic infrastructure and its application to a model of orientation selectivity. *IEEE Transactions on Circuits and Systems I*, 5(54): 981–993, 2007. doi: 10.1109/TCSI.2007.893509. URL http://ncs.ethz.ch/pubs/pdf/Chicca_etal07b.pdf.

S. Choudhary, S. Sloan, S. Fok, A. Neckar, E. Trautmann, P. Gao, T. Stewart, C. Eliasmith, and K. Boahen. Silicon neurons that compute. In A. Villa, W. Duch, P. Érdi, F. Masulli, and G. Palm, editors, *Artificial Neural Networks and Machine Learning – ICANN 2012*, volume 7552 of *Lecture Notes in Computer Science*, pages 121–128. Springer Berlin / Heidelberg, 2012. ISBN 978-3-642-33268-5. doi: 10.1007/978-3-642-33269-2_16.

C.E. Connor, H.E. Egeth, and S. Yantis. Visual attention: Bottom-up versus top-down. *Current Biology*, 14:850–852, 2004.

J. Conradt, M. Cook, R. Berner, P. Lichtsteiner, R.J. Douglas, and T. Delbruck. A pencil balancing robot using a pair of AER dynamic vision sensors. In *International Symposium on Circuits and Systems, (ISCAS), 2009*, pages 781–784. IEEE, May 2009. doi: 10.1109/ISCAS.2009.5117867.

S.M. Crouzet, H. Kirchner, and S.J. Thorpe. Fast saccades toward faces: Face detection in just 100 ms. *Journal of Vision*, 10(4), 2010. doi: 10.1167/10.4.16.

P. Dayan and L.F. Abbott. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. MIT Press, Cambridge, MA, USA, 2001. ISBN 9780262541855.

S.R. Deiss, R.J. Douglas, and A.M. Whatley. A pulse-coded communications infrastructure for neuromorphic systems. In W. Maass and C.M. Bishop, editors, *Pulsed Neural Networks*, chapter 6, pages 157–78. MIT Press, 1998.

T. Delbruck. Frame-free dynamic digital vision. In et al. (Eds.) K. Hotate, editor, *Proc. of the Intl. Symp. on Secure-Life Electronics*, volume 1, pages 21–26. University of Tokyo, 2008.

T. Delbruck and A. Van Schaik. Bias current generators with wide dynamic range. *Analog Integrated Circuits and Signal Processing*, 43(3):247–268, 2005.

A.M. Derrington and P. Lennie. Spatial and temporal contrast sensitivities of neurones in lateral geniculate nucleus of macaque. *Journal of Physiology*, 357:219–240, 1984.

M. Dick, S. Ullman, and D. Sagi. Parallel and serial processes in motion detection. *Science*, 237:400–402, 1987.

R.J. Douglas, M.A. Mahowald, and C. Mead. Neuromorphic analogue VLSI. *Annu. Rev. Neurosci.*, 18:255–281, 1995.

C. Eliasmith. How to build a brain: from function to implementation. *Synthese*, 159(3): 373–388, 2007.

C. Eliasmith and C.H. Anderson. Developing and applying a toolkit from a general neurocomputational framework. *Neurocomputing*, 26:1013–1018, 1999.

C. Eliasmith and C.H. Anderson. *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. The MIT Press, 2004.

C. Eliasmith, T.C. Stewart, X. Choo, T. Bekolay, T. DeWolf, Y. Tang, and D. Rasmussen. A large-scale model of the functioning brain. *Science*, 338(6111):1202–1205, 2012. doi: 10.1126/science.1225266. URL http://www.sciencemag.org/content/338/6111/1202.abstract.

C. Enroth-Cugell and J.G. Robson. The contrast sensitivity of retinal ganglion cells of the cat. *Journal of physiology*, 187(3):517–552, 1966.

C.W. Eriksen and J.E. Hoffman. Temporal and spatial characteristics of selective encoding from visual displays. *Perception & Psychophysics*, 12:201–204, 1972.

D.B. Fasnacht and G. Indiveri. A PCI based high-fanout AER mapper with 2 GiB RAM look-up table, 0.8 μs latency and 66 mhz output event-rate. In *Conference on Information Sciences and Systems, CISS 2011*, pages 1–6, Johns Hopkins University, March 2011. doi: 10.1109/CISS.2011.5766102. URL http://ncs.ethz.ch/pubs/pdf/Fasnacht_Indiveri11.pdf.

D.B. Fasnacht, A.M. Whatley, and G. Indiveri. A serial communication infrastructure for multi-chip address event system. In *International Symposium on Circuits and Systems, (ISCAS), 2008*, pages 648–651. IEEE, May 2008. doi: 10.1109/ISCAS.2008.4541501. URL http://ncs.ethz.ch/pubs/pdf/Fasnacht_etal08.pdf.

M.A. Freed, R.G. Smith, and P. Sterling. Computational model of the on-alpha ganglion cell receptive field based on bipolar cell circuitry. *Proceedings of the National Academy of Sciences*, 89:236–240, 1992.

S. Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search*. PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, January 2006.

S. Frintrop, M. Klodt, and E. Rome. A real-time visual attention system using integral images. In *Proceedings of the 5th international conference on computer vision systems*, 2007.

S. Furber and S. Temple. Neural systems engineering. *Journal of the Royal Society interface*, 4(13):193–206, 2007.

P.E. Gronowski, W.J. Bowhill, R.P. Preston, M.K. Gowan, and R.L. Allmon. High-performance microprocessor design. *Solid-State Circuits, IEEE Journal of*, 33(5):676–686, 1998.

D. Heinke and G.W. Humphreys. Computational models of visual selective attention: A review. *Connectionist models in cognitive psychology*, 1(4):273–312, 2005.

G. Indiveri. Neuromorphic analog VLSI sensor for visual tracking: Circuits and application examples. *IEEE Transactions on Circuits and Systems II*, 46(11):1337–1347, 1999. doi: 10.1109/82.803473. URL http://ncs.ethz.ch/pubs/pdf/Indiveri99.pdf.

G. Indiveri. A 2D neuromorphic VLSI architecture for modeling selective attention. In S.-I. Amari, C.L. Giles, M. Gori, and V. Piuri, editors, *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks; IJCNN 2000*, volume IV, pages 208–213. IEEE Computer Society, 2000a. doi: 10.1109/IJCNN.2000.860774. URL http://ncs.ethz.ch/pubs/pdf/Indiveri00.pdf.

G. Indiveri. Modeling selective attention using a neuromorphic analog VLSI device. *Neural Computation*, 12(12):2857–2880, 2000b. doi: doi:10.1162/089976600300014755. URL http://ncs.ethz.ch/pubs/pdf/Indiveri00b.pdf.

G. Indiveri. A current-mode hysteretic winner-take-all network, with excitatory and inhibitory coupling. *Analog Integrated Circuits and Signal Processing*, 28(3):279–291, September 2001. URL http://ncs.ethz.ch/pubs/pdf/Indiveri01.pdf.

G. Indiveri and E. Chicca. A VLSI neuromorphic device for implementing spike-based neural networks. In *Neural Nets WIRN11 - Proceedings of the 21st Italian Workshop on Neural Nets*, pages 305–316, Jun 2011. ISBN 978-1-60750-971-4. doi: 10.3233/978-1-60750-972-1-305. URL http://ncs.ethz.ch/pubs/pdf/Indiveri_Chicca11.pdf.

G. Indiveri, E. Chicca, and R.J. Douglas. A VLSI array of low-power spiking neurons and bistable synapses with spike–timing dependent plasticity. *IEEE Transactions on Neural Networks*, 17(1):211–221, Jan 2006. doi: 10.1109/TNN.2005.860850. URL http://ncs.ethz.ch/pubs/pdf/Indiveri_etal06.pdf.

Giacomo Indiveri and Timothy K Horiuchi. Frontiers in neuromorphic engineering. *Frontiers in Neuroscience*, 5(118), 2011. ISSN 1662-453X. doi: 10.3389/fnins.2011.00118. URL http://www.frontiersin.org/neuromorphic_engineering/10.3389/fnins.2011.00118/full.

L. Itti. Real-time high-performance attention focusing in outdoors color video streams. In Bernice E. Rogowitz and Thrasyvoulos N. Pappas, editors, *Human Vision and Electronic Imaging VII*, volume 4662, pages 235–243. SPIE, 2002. URL http://link.aip.org/link/?PSI/4662/235/1.

L. Itti. The iLab neuromorphic vision C++ toolkit: Free tools for the next generation of vision algorithms. *The Neuromorphic Engineer*, 1(1):10, Mar 2004.

L. Itti. Quantifying the contribution of low-level saliency to human eye movements in dynamic scenes. *Visual Cognition*, 12(6):1093–1123, 2005.

L. Itti and C. Koch. Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, 2001a.

L. Itti and C. Koch. Feature combination strategies for saliency-based visual attention systems. *Journal of Electronic Imaging*, 10(1):161–169, 2001b. URL http://link.aip.org/link/?JEI/10/161/1.

L. Itti, E. Niebur, and C. Koch. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

E.I. Knudsen and M. Konishi. Center-surround organization of auditory receptive fields in the owl. *Science*, 202(4369):778–780, November 1978.

C. Koch and S Ullman. Shifts in selective visual-attention – towards the underlying neural circuitry. *Human Neurobiology*, 4(4):219–227, 1985.

S.M. Kosslyn, R.A. Flynn, J.B. Amsterdam, and G. Wang. Components of high-level vision: A cognitive neuroscience analysis and accounts of neurological syndromes. *Cognition*, 34 (3):203–277, 1990.

S.W. Kuffler. Discharge patters and functional organization of mammalian retina. *Journal of Neurophysiology*, 16:37–68, 1953.

M.F. Land and R.D. Fernald. The evolution of eyes. *Annual review of neuroscience*, 15(1): 1–29, 1992.

A.G. Leventhal, editor. *The Neural Basis of Visual Function (Vision and Visual Dysfunction Vol 4)*. The Macmillan Press, 1991.

P. Lichtsteiner, C. Posch, and T. Delbruck. An 128x128 120dB 15μs-latency temporal contrast vision sensor. *IEEE J. Solid State Circuits*, 43(2):566–576, 2008.

R.A. Linsenmeier, L.J. Frishman, H.G. Jakiela, and C. Enroth-Cugell. Receptive field properties of X and Y cells in the cat retina derived from contrast sensitivity measurements. *Vision Research*, 22:1173–1183, 1982.

C.A. Mack. Fifty years of moore's law. *Semiconductor Manufacturing, IEEE Transactions on*, 24(2):202–207, 2011.

M.A. Mahowald. *VLSI analogs of neuronal visual processing: a synthesis of form and function*. PhD thesis, Department of Computation and Neural Systems, California Institute of Technology, Pasadena, CA., 1992.

S. Martinez-Conde, S.L. Macknik, and D.H. Hubel. The role of fixational eye movements in visual perception. *Nature Reviews Neuroscience*, 5(3):229–240, March 2004. doi: 10.1038/nrn1348.

S. Martinez-Conde, S.L. Macknik, X.G. Troncoso, and T.A. Dyar. Microsaccades counteract visual fading during fixation. *Neuron*, 49(2):297–305, January 2006.

S. May, M. Klodt, E. Rome, and R. Breithaupt. GPU-accelerated affordance cueing based on visual attention. In *International Conference on Intelligent Robots and Systems*, pages 3385–3390. IEEE, 2007.

P. McLeod, J. Driver, and J. Crisp. Visual search for a conjunction of movement and form is parallel. *Nature*, 332:154–155, 1988.

C.A. Mead. *Analog VLSI and Neural Systems*. Addison-Wesley, Reading, MA, 1989.

G.E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8):114–117, 1965.

T.G. Morris and S.P. DeWeerth. Analog VLSI excitatory feedback circuits for attentional shifts and tracking. *Analog Integrated Circuits and Signal Processing*, 13(1/2):79–92, May/June 1997.

T.G. Morris, T.K. Horiuchi, and S.P. DeWeerth. Object-based selection within an analog VLSI visual attention system. *IEEE Transactions on Circuits and Systems II*, 45(12):1564–1572, 1998.

K. Nakayama and G.H. Silverman. Serial and parallel processing of visual feature conjunctions. *Nature*, 320:154–155, 1986.

E. Neftci and G. Indiveri. A device mismatch compensation method for VLSI spiking neural networks. In *Biomedical Circuits and Systems Conference (BioCAS), 2010*, pages 262–265. IEEE, 2010. doi: 10.1109/BIOCAS.2010.5709621. URL http://ncs.ethz.ch/pubs/pdf/Neftci_Indiveri10.pdf.

E. Neftci, E. Chicca, G. Indiveri, and R.J. Douglas. A systematic method for configuring VLSI networks of spiking neurons. *Neural Computation*, 23(10):2457–2497, Oct. 2011. doi: 10.1162/NECO_a_00182. URL http://ncs.ethz.ch/pubs/pdf/Neftci_etal11.pdf.

U. Neisser. *Cognitive psychology*, volume 4. Appleton-Century-Crofts New York, 1967.

N. Ouerhani and H. Hügli. Real-time visual attention on a massively parallel simd architecture. *Real-Time Imaging*, 9(3):189–196, 2003.

N. Ouerhani, H. Hügli, P. Burgi, and P. Ruedi. A real time implementation of the saliency-based model of visual attention on a simd architecture. In L. Van Gool, editor, *Lecture notes in computer science. Pattern Recognition: 24th DAGM Symposium, Zurich, Switzerland*, volume 2449/2002, pages 282–289. Springer Berlin / Heidelberg, September 2002.

E. Painkras, L.A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D.R. Lester, A.D. Brown, and S.B. Furber. SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation. *IEEE Journal of Solid-State Circuits*, 48(8):–, August 2013. ISSN 0018-9200. doi: 10.1109/JSSC.2013.2259038.

M.I. Posner, C.R.R. Snyder, and B.J.Davidson. Attention and the detection of signals. *Journal of Experimental Psychology*, 109(2):160–174, 1980.

R.A. Rensink, J.K. O'Regan, and J.J. Clark. To see or not to see: The need for attention to perceive changes in scenes. *Psychological Science*, 8(5):368–373, 1997.

I.H. Robertson and J.M.J. Murre. Rehabilitation of brain damage: Brain plasticity and principles of guided recovery. *Psychological bulletin*, 125(5):544, 1999.

R.W. Rodieck. Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vision Research*, 5:583–601, 1965.

R.W. Rodieck and J. Stone. Response of cat retinal ganglion cells to moving visual patterns. *Journal of Neurophysiology*, 28:819–832, 1965a.

R.W. Rodieck and J. Stone. Analysis of receptive fields of cat retinal ganglion cells. *Journal of Neurophysiology*, 28:833–849, 1965b.

C.S. Royden, J.M. Wolfe, and N. Klempen. Visual search asymmetries in motion and optic flow fields. *Perception & Psychophysics*, 63:436–444, 2001.

P.-F. Ruedi, P.R. Marchal, and X. Arreguit. A mixed digital-analog SIMD chip tailored for image perception. In *International Conference on Image Processing*, volume 1, pages 1011–1014. IEEE, 1996.

S. Sheik, E. Chicca, and G. Indiveri. Exploiting device mismatch in neuromorphic VLSI systems to implement axonal delays. In *International Joint Conference on Neural Networks, IJCNN 2012*, pages 1940–1945. IEEE, 2012. URL http://ncs.ethz.ch/pubs/pdf/Sheik_etal12b.pdf.

R. Silver, K. Boahen, S. Grillner, N. Kopell, and K.L. Olsen. Neurotech for neuroscience: unifying concepts, organizing principles, and emerging tools. *Journal of Neuroscience*, 27 (44):11807, 2007.

M. Sivilotti. *Wiring considerations in analog VLSI systems with application to field-programmable networks*. PhD thesis, Comput. Sci. Div., California Inst. Technol., Pasadena, CA, 1991.

D. Sonnleithner and G. Indiveri. A real-time event-based selective attention system for active vision. In *Advances in Autonomous Mini Robots*, pages 205–219. Springer Berlin Heidelberg, May 2012. doi: 10.1007/978-3-642-27482-4_21. URL http://ncs.ethz.ch/pubs/pdf/Sonnleithner_Indiveri12.pdf.

A.D. Straw. Vision Egg: An open-source library for realtime visual stimulus generation. *Frontiers in Neuroinformatics*, 2, 2008. ISSN ISSN 1662-5196. doi: 10.3389/neuro.11/004.2008. URL frontiersin.org/neuroinformatics/paper/10.3389/neuro.11/004.2008/.

A.M. Treisman and G. Gelade. A feature–integration theory of attention. *Cognit Psychol*, 12: 97–136, 1980.

J.K. Tsotsos. Computational foundations for attentive processes. In L. Itti, G. Rees, and J.K. Tsotsos, editors, *Neurobiology of Attention*, pages 3–7. Elsevier, 2005.

Guido van Rossum. Python reference manual. CWI Report, May 1995.

John von Neumann. First draft of a report on the edvac. *IEEE Ann. Hist. Comput.*, 15(4): 27–75, 1993. ISSN 1058-6180. doi: 10.1109/85.238389.

D. Vucinic, L.B. Cohen, and E.K. Kosmidis. Interglomerular center-surround inhibition shapes odorant-evoked input to the mouse olfactory bulb in vivo. *Journal of Neurophysiology*, 95(3):1881–1887, 2006. doi: 10.1152/jn.00918.2005.

G.L. Walls. The vertebrate retina and its adaptive radiation. *Psychology*, 45:205–217, 1942.

D. Walther. *Interactions of visual attention and object recognition: computational modeling, algorithms, and psychophysics*. PhD thesis, California Institute of Technology, 2006.

D. Walther and C. Koch. Modeling attention to salient proto-objects. *Neural Networks*, 19: 1395–1407, 2006. doi: 10.1016/j.neunet.2006.10.001.

C.S. Wilson, T.G. Morris, and S.P. DeWeerth. A two-dimensional, object-based analog VLSI visual attention system. In S.P. DeWeerth, S.M. Wills, and A.T. Ishii, editors, *Proceedings of the 20th Anniversary Conference on Advanced Research in VLSI*, volume 20, pages 291–308. IEEE Computer Society Press, 1999. Los Alamitos, CA.

J.M. Wolfe. Guided search 2.0 a revised model of visual search. *Psychonomic bullettin & review*, 1(2):202–238, 1994.

J.M. Wolfe. Guided search 4.0: A guided search model that does not require memory for rejected distractors. *Journal of Vision*, 1(3):349–349, 2001.

J.M. Wolfe. Guided search 4.0. *Integrated models of cognitive systems*, pages 99–120, 2006.

J.M. Wolfe and G. Gancarz. Guided search 3.0. In V. Lakshminarayanan, editor, *Basic and clinical applications of vision science*, volume 60 of *Documenta Ophthalmologica Proceedings Series*, pages 189–192. Springer Netherlands, 1997. doi: 10.1007/978-94-011-5698-1_30.

J.M. Wolfe and T.S. Horowitz. What attributes guide the deployment of visual attention and how do they do it? *Nature Review Neuroscience*, 5(4):495–501, June 2004.

J.M. Wolfe, K.R. Cave, and S.L. Franzel. Guided search: An alternative to the feature integration model for visual search. *Journal of Experimental Psychology*, 15(3):419–433, 1989.

T. Xu, T. Pototschnig, K. Kuhnlenz, and M. Buss. A high-speed multi-GPU implementation of bottom-up attention using CUDA. In *International Conference on Robotics and Automation*, pages 41–47. IEEE, 2009.

S. Yantis and H.E. Egeth. On the distinction between visual salience and stimulus-driven attentional capture. *Journal of Experimental Psychology*, 25:661–676, 1999.

C. Zhang and Z.Y. Zhang. A survey of recent advances in face detection. Technical report, Microsoft Research, 2010.

A AMDA board firmware

The AER Motherboard with D/A converters (AMDA) board is used to control the input bias voltages for a variety of neuromorphic chips. It contains a microcontroller that enables the communication with the workstation and controls the Digital–to–Analog Converters (DACs) and Analog–to–Digital Converters (ADCs) on the board. In this chapter I describe the structure of the microcontroller's firmware and the tools to upload the firmware to the microcontroller on the AMDA board.

A.1 Firmware structure

The AMDA board is equipped with an Atmel ATmega128 microcontroller. It is based on a 8-bit RISC architecture [ATmega128]. The firmware is written in C.

As shown in Fig.A.1 the firmware is divided into two parts: One part takes care of the correct communication with the DACs, ADCs, the LED and the USART interface. Program code for higher levels is pooled in the logic part. This high level code makes use of the routines provided by the IO part. Tables A.1 and A.2 give an overview of the purpose of the different C files.

The main routine defined in the file `main.c` has three objectives:

1. When the board is connected to a 5 V power source the microcontroller boots up. The main routine ensures that all devices are initialized.

2. The main routine polls the USART input buffer. If the buffer contains a character string the main parser is called to identify the command and eventually execute it.

File	Description
`aIO.*`	Basic IO functions to control the DACs and the ADCs on the AMDA board. To apply a voltage to one of the output pins, you first have to store the desired value(s) using the `storeAnalogOutPin`-function and then call the `applyAnalogPins`-function.
`dIO.*`	Basic IO functions for the digital output and input pins.
`led.*`	Contains functions to control the LED on the AMDA board.
`usartIO.*`	Provides functions to communicate with the workstation via USART (serial interface RS232). The firmware on the AMDA board communicates with 230 400 baud with 8-bit characters.

Table A.1: Description of the content of the code files in the IO part.

File	Description
`analog.*`	Provides the parser for all commands to control either the DACs or ADCs. The parser functions call for each command a subfunction that is implemented as internal function. This function calls the IO functions in `aIO.h`. If you want to implement additional commands for the analog devices on the board these should go in these files, whereas additional analog IO functionalities should be added to `aIO.h`.
`biasGenerator.*`	Provides the parser for the bias generator functions. The bias generator uses mainly the digital IO functions provided by `dIO.h`. Port DOut1 is used for the bias generator's clock Port DOut2 is used for the bias generator's input Port DOut3 is used for the latch signal To avoid conflicts with chips not using a bias generator with the given port map, the firmware checks if pin DIn1 is set to high. If not, the firmware does not execute bias generator commands.
`board.*`	Provides the parser for the board functions. These are functions to set the LED mode, to define the board's ID, to restart the board's firmware, to get the current temperature, to get the fimrware's version, and to print a usage text on the USART console.
`digital.*`	Provides the parser for the digital functions. These functions set the digital pins and get the value of the digital input pin.
`main.c`	Contains the main routine of the AMDA firmware and the error handling functionality.
`parser.*`	Provides the main parser. The commands have a prefix for each module, such as analog (`A.*`), digital (`D.*`), bias generator (`BG.*`) and board (`B.*`). The main parser sorts the incoming commands according to these prefixes and calls the module's parser. If a new module should be developed the call of the module parser should be inserted in this files.
`utilities.*`	Provides utilities function such as a sleep function of a pin conversion function.
`usage.txt`	Contains the usage text that is display if the command `B.SHU` is sent to the AMDA board's firmware. Should be updated if new commands are defined.

Table A.2: Description of the content of the code files in the logic part.

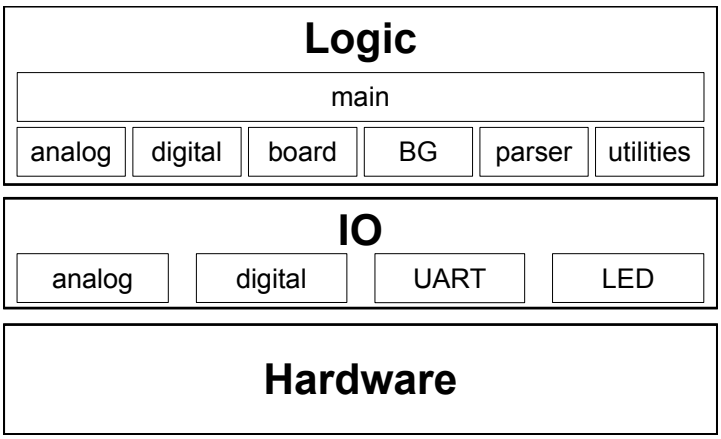


Figure A.1: Schematic of the logical structure of the AMDA board firmware. The software is split into low level (IO) and high level (Logic) parts.

3. Whenever an error occurs in the firmware a negative integer value is return to the main method. The main method then looks up the corresponding error message and transmits it via the USART interface to the workstation.

A.2 Programming the AMDA board's microcontroller

To program the AMDA board's microcontroller the board has to be connected via the JTAG interface with the help of a programmer (e.g. the Atmel AVRISP mkII) to the (Linux) workstation. The program to communicate with the programmer is called `avrdude`. First the ATmega128's fuse and lock bits have to be programmed. Therefore the command

```
avrdude -c AVRISPMKII -p m128 -P usb -U lfuse:w:lfuse.hex -U hfuse:w:hfuse.hex
                    -U efuse:w:efuse.hex -U lock:w:lock.hex
```

has to be executed. It is only necessary to call this command once. After the fuse and lock bits are set the firmware has to be uploaded to the microcontroller's flash memory. The command is:

```
avrdude -c AVRISPMKII -p m128 -P usb -U flash:w:HEXFILE
```

`HEXFILE` has to be replaced by the name of the hex-file containing the firmware.

B Abbreviations

ADC Analog–to–Digital Converter

AER Address Event Representation

AEX AMDA EXtension board

AMDA AER Motherboard with D/A converters

ALU Arithmetic Logic Unit

CAD Computer Aided Design

cDVS Color DVS

CMOS Complementary Metal–Oxide–Semiconductor

CPU Central Processing Unit

CPLD Complex Programmable Logic Device

DAC Digital–to–Analog Converter

DMA Direct Memory Access

DVS Dynamic Vision Sensor

FPGA Field Programmable Gate Array

GPU Graphical Processing Unit

GUI Graphical User Interface

IF2DWTA Integrate & Fire 2–Dimensional WTA

I&F Integrate & Fire

INI Institute of Neuroinformatics

IO Input / Output

IOR Inhibition of Return

JTAG Joint Test Action Group

LGN Lateral Geniculate Nucleus

NEF Neural Engineering Framework

pAER parallel AER

PCB Printed Circuit Board

PCI Peripheral Component Interconnect

RISC Reduced Instruction Set Computer

SAC Selective Attention Chip

sAER serial AER

SIMD Single Instruction Multiple Data

SPI Serial Peripheral Interconnect

USART Universal Synchronous/Asynchronous Receiver Transmitter

USB Universal Serial Bus

VLSI Very Large Scale Integration

WTA Winner–Take–All