

# Analog VLSI Implementations of Visual Motion Sensors and a Neuromorphic Obstacle Avoidance System

by  
Erhan Özalevli

A Thesis Submitted to the Faculty of the  
ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT  
In Partial Fulfillment of the Requirements  
For the Degree of  
MASTER OF SCIENCE



THE UNIVERSITY OF ARIZONA  
Tucson, Arizona

2003  
(Submitted June 4, 2003)

## STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the copyright holder.

SIGNED: \_\_\_\_\_  
ERHAN ÖZALEVLI

Approval by Thesis Director  
This thesis has been approved on the date shown below:

---

Charles M. Higgins  
Assistant Professor of Electrical and Computer Engineering

---

Date

© 2003  
by  
Erhan Özalevli  
All Rights Reserved

# DEDICATION

To My Family

# Acknowledgements

I would like to thank my family for their endless support and love through all my endeavors.

My gratitude goes to my advisor Prof. Higgins. I am very indebted to him for his encouragement, support and guidance throughout all my studies and research. Most importantly I am very thankful for his generous efforts to provide me with the freedom to imagine.

I am thankful to Prof. Palusinski and Prof. Rozelblit for serving on my thesis defense committee.

Also, I would like to thank all the members of the Higgins Lab for creating a pleasant atmosphere to work in. I am thankful to Sam for his kindness and help in my first year and to Vivek, Anusha, Jad and Robert for their good company. I am grateful to Jad for his efforts to make me smile even at my most stressful moments. Besides, I am thankful to Anusha and Jad for their help in the manuscript of the thesis.

Lastly, I would like to thank to my friends Aytac, Aycan, Serdar, Zafer, Koray, Yakup and Gunay for their friendship and good company...

# Contents

<b>Acknowledgements</b>	<b>iv</b>
<b>Abstract</b>	<b>1</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Motion detection algorithms and implementations . . . . .	3
1.2 Monolithic and multi-chip implementations . . . . .	4
1.3 Obstacle avoidance . . . . .	5
1.4 Organization of the thesis . . . . .	5
<b>2 Analog VLSI building blocks</b>	<b>6</b>
2.1 MOSFET operation . . . . .	6
2.2 Adaptive photoreceptor . . . . .	8
2.3 Differential transconductance amplifier . . . . .	10
2.4 Full-wave rectifier and current comparator . . . . .	12
2.5 Absolute-value and squaring circuits . . . . .	13
2.6 Nonlinear differentiator . . . . .	15
2.7 Winner-take-all circuit . . . . .	17
2.8 One and four-quadrant multipliers . . . . .	20
<b>3 Monolithic Implementations of Analog VLSI Motion Sensors</b>	<b>22</b>
3.1 Hysteretic winner-take-all based motion sensor . . . . .	22
3.1.1 Algorithm of the sensor . . . . .	23
3.1.2 Transistor-level implementation of the algorithm . . . . .	26
3.2 Nonlinear differentiator based direction sensor . . . . .	35
3.2.1 Algorithm of the sensor . . . . .	35
3.2.2 Transistor-level implementation of the algorithm . . . . .	38
<b>4 Multi-chip implementations of Analog VLSI Motion Sensors</b>	<b>44</b>
4.1 AER protocol . . . . .	44
4.2 Photosensitive sender chip . . . . .	45
4.3 Motion computing receiver chips . . . . .	47
4.3.1 Hassenstein-Reichardt model . . . . .	53
4.3.1.1 Modified form: . . . . .	55
4.3.1.2 Expected response: . . . . .	55
4.3.1.3 Multi-chip implementation of the model: . . . . .	56
4.3.2 Adelson-Bergen spatiotemporal energy model . . . . .	57
4.3.2.1 Canonical Form: . . . . .	59
4.3.2.2 Modified Form: . . . . .	61
4.3.2.3 Expected Response: . . . . .	61
4.3.2.4 Multi-chip implementation of the model: . . . . .	62
4.3.3 Barlow-Levick model . . . . .	63
4.3.3.1 Canonical form: . . . . .	66

4.3.3.2	Modified form: . . . . .	66
4.3.3.3	Expected response: . . . . .	66
4.3.3.4	Multi-chip implementation of the model: . . . . .	66
4.4	Characterization results . . . . .	70
<b>5</b>	<b>Obstacle Avoidance Algorithm</b>	<b>79</b>
5.1	Algorithm . . . . .	80
5.1.1	Motion Parallax . . . . .	80
5.1.2	The Centering Behavior . . . . .	82
5.1.3	Looming Sensitive Neurons and Time-To-Collision Computation . . . . .	86
5.2	Hardware Architecture of the Centering Behavior . . . . .	87
5.2.1	Circuit implementation of the direction insensitive movement detector neuron	92
5.3	Hardware Architecture of the Escape Behavior . . . . .	95
5.4	Algorithm Implemented on an Agent . . . . .	96
<b>6</b>	<b>Discussion</b>	<b>100</b>
	<b>Bibliography</b>	<b>102</b>
<b>A</b>	<b>Chip Layouts</b>	<b>107</b>
<b>B</b>	<b>Chip biases</b>	<b>127</b>
B.1	Sender chip biases . . . . .	128
B.2	Adelson-Bergen receiver chip biases . . . . .	129
B.3	Hassenstein-Reichardt receiver chip biases . . . . .	130
B.4	Barlow-Levick receiver chip biases . . . . .	131

# List of Figures

2.1	Cross section of the MOSFET and $I_{ds}$ versus $V_{gs}$ curve of an n-type MOSFET. . . . .	7
2.2	The adaptive photoreceptor circuit and its response to intensity and frequency changes . . . . .	8
2.3	The transconductance amplifier circuit and its I-V curve . . . . .	10
2.4	The full wave rectifier and current comparator circuits . . . . .	13
2.5	The full wave rectifier and normalized squaring circuit. . . . .	14
2.6	The nonlinear differentiator circuit. . . . .	16
2.7	Current output of the nonlinear differentiator circuit to a 10Hz stimulus. . . . .	18
2.8	Schematics of the winner-take-all and hysteretic winner-take-all circuits and experimental data for a two-input winner-take-all circuit. . . . .	19
2.9	One and four-quadrant current mode multipliers. . . . .	21
3.1	The Barlow and Levick model of direction selectivity. . . . .	23
3.2	Hysteretic winner-take-all based elementary motion detector. . . . .	24
3.3	Functioning of the hysteretic winner-take-all based motion detection algorithm in signal level. . . . .	25
3.4	The first version of the transistor-level implementation of the hysteretic winner-take-all based motion algorithm. . . . .	27
3.5	Response of two half wave rectifiers, which are used to rectify the responses of transconductance amplifiers to the sinusoidal 50Hz gratings with 90-degree phase difference. . . . .	30
3.6	Responses of hysteretic winner-take-all circuit . . . . .	30
3.7	The output of the first version of the circuit is obtained by sampling the loser of the competition by the winner output. . . . .	31
3.8	The second version of the circuit implementation of the hysteretic winner-take-all based motion algorithm. . . . .	32
3.9	The final output and the computation performed by the second version of the circuit for 10Hz sinusoidal grating input. . . . .	33
3.10	The final output of the hysteretic winner-take-all circuit . . . . .	34
3.11	The algorithm of nonlinear differentiator based direction sensor . . . . .	35
3.12	The result of Equation 3.24 when a normalized sinusoidal grating with 45-degree phase difference is used to simulate preferred direction, no-motion and null direction cases respectively. . . . .	37
3.13	The circuit implementation of the nonlinear differentiator based motion algorithm. . . . .	40
3.14	The response of the first version of nonlinear differentiator based algorithm. . . . .	41
3.15	The second version of the circuit implementation of the nonlinear differentiator based motion sensor. . . . .	42
3.16	The response of the second version of the circuit implementation of the nonlinear differentiator based motion sensor to a 10Hz sinusoidal grating. . . . .	43
4.1	The organization of P and N pixels in the sender and receiver chips. . . . .	45
4.2	AER protocol summary . . . . .	46
4.3	AER interface circuit located in the sender chip. . . . .	47
4.4	Sender chip pixel circuitry. . . . .	48
4.5	The second version of the sender chip pixel circuitry. . . . .	49

4.6	AER circuit in the receiver chip. . . . .	50
4.7	The current outputs of two integrators biased with different $V_{qua}$ and $V_{off}$ biases. . .	52
4.8	The Reichardt model. . . . .	54
4.9	Modified version of the Reichardt model. . . . .	54
4.10	A multi-chip implementation of the Reichardt model. . . . .	57
4.11	The Reichardt pixel circuitry. . . . .	58
4.12	The structure of the Adelson-Bergen model in terms of linear filters . . . . .	59
4.13	The original form of the Adelson-Bergen model . . . . .	60
4.14	Modified version of the Adelson-Bergen spatiotemporal energy model. . . . .	62
4.15	The multi-chip implementation of the Adelson-Bergen algorithm. . . . .	63
4.16	The P pixel in the Adelson-Bergen receiver chip. . . . .	64
4.17	The second version of the P-pixel in the Adelson-Bergen receiver chip. . . . .	65
4.18	Modified version of the Barlow and Levick model. . . . .	67
4.19	Multi-chip implementation of the Barlow and Levick model. . . . .	67
4.20	The P pixel in the Barlow-Levick receiver chip. . . . .	68
4.21	The structure of the main pixel in the Barlow-Levick receiver chip. . . . .	69
4.22	Raw data obtained from the motion sensors . . . . .	72
4.23	Responses of the sensors to stimulus with different orientations . . . . .	73
4.24	Responses of the sensors to change in the contrast level. . . . .	74
4.25	Temporal frequency sweep responses of sensors . . . . .	75
4.26	Variation of temporal frequency tuning of sensors. . . . .	76
4.27	Spatial frequency sweep responses of sensors . . . . .	77
4.28	Spatiotemporal frequency plots of sensors . . . . .	78
5.1	An illustration of motion parallax . . . . .	80
5.2	Motion contrast . . . . .	84
5.3	Navigation achieved by using different centering techniques. . . . .	85
5.4	Multiple-correlator scheme . . . . .	88
5.5	The modular implementation of the multi-correlator based velocity sensor. . . . .	89
5.6	An implementation of multi-correlator based velocity sensor. . . . .	90
5.7	The second implementation of multi-correlator based velocity sensor. . . . .	91
5.8	The circuit implementation of the P and N pixels in the receiver chip . . . . .	94
5.9	The final implementation of the escape response. . . . .	97
5.10	The final implementation of the centering response and the escape response on an agent. . . . .	99
A.1	The layout of the first version of the pixel level implementation of the hysteretic winner-take-all based motion sensor. . . . .	108
A.2	The layout of the second version of the pixel of the hysteretic winner-take-all based motion sensor. . . . .	109
A.3	The final chip layout of the hysteretic winner-take-all based motion sensor. This visual motion sensor is called WTA. . . . .	110
A.4	The layout of the first version of the nonlinear differentiator based motion sensor. . .	111
A.5	The layout of the second version of the nonlinear differentiator based motion sensor. .	112
A.6	The layout of the third version of the nonlinear differentiator based motion sensor. . .	113
A.7	The final chip layout of the nonlinear differentiator based motion sensor. This chip's name is NLD. . . . .	114
A.8	The layout of sender pixel used for multi-chip implementations. . . . .	115
A.9	The final layout of the sender chip. The name of this chip is ABSndr3. . . . .	116
A.10	The receiver pixel layout of the Reichardt correlation model. . . . .	117
A.11	The receiver chip layout that implements the Reichardt correlation model. The chip name is Reichardt_rcvr2. . . . .	118
A.12	The pixel layout of the first version of the Adelson-Bergen receiver chip. . . . .	119
A.13	The first version of the receiver chip layout that realizes the Adelson-Bergen algorithm. The chip name is ABmrcvr3. . . . .	120

A.14	The pixel layout of the second version of the Adelson-Bergen receiver chip. . . . .	121
A.15	The second version of the receiver chip layout that realizes the Adelson-Bergen algorithm. This chip is called ABmrcvr3a. . . . .	122
A.16	The layout of receiver pixel that implements the Barlow-Levick model. . . . .	123
A.17	The receiver chip layout that implements the Barlow-Levick model. The chip name is BL_rcvr. . . . .	124
A.18	The layout of the centering response receiver pixel. . . . .	125
A.19	The final layout implementation of the centering response receiver chip. This chip is called ObsavABrcvr. . . . .	126

# Abstract

Visual motion perception plays a vital role in the process of behavioral computations that are performed during the interaction of organisms with their environment. In the early stages of motion detection, visual information is processed by a large number of elementary motion detectors to obtain a representation of the visual field in terms of local motion vectors. In this work, we present analog VLSI implementations of motion detection algorithms that are based not only on biological models but also on the computational properties of motion perception. First, we describe monolithic implementations of hysteretic winner-take-all and nonlinear-differentiator based algorithms. These compact elementary motion detector models can reliably be used to obtain high resolution sensors. Second, we explain multi-chip implementations of biomimetic intensity-based models, namely Adelson-Bergen, Hassenstein-Reichardt, and Barlow-Levick models. By employing a modular strategy, these algorithms are successfully implemented without much sacrifice of the fill factor in the front-end chip. In addition, we describe an obstacle avoidance algorithm that is realized by incorporating a multi-chip version of the Adelson-Bergen algorithm with centering behavior and time-to-collision computation. In this way, the overall system can successfully acquire clues about the structure of its environment so that collisions can be effectively avoided. This system might be employed in building a robot that can navigate in complex cluttered environments.

# Chapter 1

## Introduction

The visual system provides a wealth of information that has genuine behavioral importance for organisms in that it enables them to survive within their environment and achieve evolutionary success. Although organisms have different perceived truth of their habitats they are anatomically and physiologically specialized in obtaining different visual clues provided by their environment. They adopt a variety of biological strategies in dealing with complex and dynamic visual tasks. Within these strategies, motion detection serves the most important function that is ubiquitously employed in organisms to process visual information. Flies employ motion information in their behavioral tasks such as gaze control, flight stabilization, and tracking [22].

As with biological systems, perception of motion information occupies a vital role in behavioral tasks achieved by artificial systems. Despite its numerous shortcomings, perceived motion virtually never fails to provide information about the outside world that has genuine behavioral importance [21]. It is utilized for a variety of purposes including tracking, collision avoidance, object recognition, time-to-flight computation, guidance, balance, and postural control.

In building biologically inspired architectures, it always has to be taken into consideration that biological models are by their nature continuous-time systems and they operate by employing a massively parallel processing strategy. These biological principles can be efficiently utilized in visual computations by employing an implementation technology that allows designers to realize layered parallel structures and power- and space-efficient architectures. In contrast to other sensory computations, optic flow computation is a very intensive process that is constrained by power consumption, and employing such technology definitely improves the performance of built systems. Conventional processors that are discrete-time and serial in operation fail to meet these criteria and are therefore not well-suited for building biologically inspired visual architectures. In addition, conventional design approaches in image processing that are implemented by employing a CCD camera together with a DSP processor cause problems such as an image transfer bottleneck, temporal aliasing and high power consumption. These systems work at high frequencies to deliver image information in a timely way and in achieving this they have to consume considerable amounts of power. Whereas, an integrative approach by analog VLSI technology and neuromorphic design principles makes intensive visual computations possible to be realized in power and space efficient systems. In contrast to conventional designs, these systems are data driven, that is, the output is sampled when there is a demand. Hence they consume little power and yield temporal aliasing-free computation.

Inspired by biological models in the visual pathways of organisms and by computational properties of visual motion perception, a variety of visual motion sensors, which consume little power and work in real time, have been developed to solve problems faced in optical flow computation. Although these sensors are built to process motion information in different ways, they all have to meet a certain number of requirements determined by computational principles of motion detection [11].

- **Two visual inputs:** Motion is a vector that is represented by two points in the visual field and therefore needs two input channels for its motion processing. For instance, input from a

single visual sampling point does not suffice to distinguish a dark bar that crossed its receptive field from the left to the right from one that crossed from the right to the left.

- **Non-linear interaction:** In order to preserve the temporal sequence information obtained from input signals, a nonlinear interaction must be incorporated into motion processing stages. Otherwise the time-averaged output of a motion detector will be equal to its response to its time-averaged input signals. Therefore, motion processing with a linear interaction between input channels cannot be directionally selective in the mean.
- **Asymmetry:** The input signals sensed by two different input channels of a motion detector have to be processed in a slightly different way. Otherwise the input channels could be interchanged without affecting the motion output. It is then no longer possible to discriminate which channel was excited first and which later. Accordingly, the detector would not be directionally selective.

## 1.1 Motion detection algorithms and implementations

Motion computation methods can be categorized into two classes, namely, feature-tracking and intensity-based algorithms.

Feature tracking algorithms detect temporal and spatial features and track them to compute the optical flow field. Evidence obtained from recordings of neurons in the visual pathway of different organisms has proved the existence of feature detectors [5], [52]. It has been verified that these neurons require some specific pattern of excitation. Based on this evidence, a variety of feature tracking algorithms have been developed. These algorithms can be classified into spatial and temporal feature tracking algorithms.

Spatial feature tracking algorithms are designed to utilize the spatial information in image sequences to compute the optical flow field. Spatial features like edges are tracked in an image sequence and based on the attained correspondence between the image sequences, the velocity of the feature can be computed [3], [57]. This algorithm is especially popular in software-based implementations because of the discrete nature of processing in feature tracking. Etienne-Cummings et al. [24] implemented this algorithm in hardware with a sensor that computes the optical flow field by tracking the appearance and disappearance of edges at neighboring pixels on its focal plane. Also, Indiveri implemented a monolithic vision chip that detects and tracks the position of the feature with highest spatial contrast [41].

In contrast to spatial feature tracking algorithms, temporal feature tracking algorithms utilize intensity changes in the optical image to compute its motion. Hardware implementations of these algorithms generally employ temporal edge detectors in order to obtain spike/pulse like responses to sudden changes in the intensity level of the image. Kramer [46] implemented a temporal feature tracking sensor by employing the FTI (facilitate, trigger and inhibit) algorithm that calculates the time of travel by utilizing the interaction between three adjacent pixels. Higgins et al. [34] demonstrated two vision sensors based on the ITI (inhibit, trigger and inhibit) and the FTC (facilitate, trigger and compare) algorithms. Moreover, a variety of velocity sensors have been developed by using the FS (facilitate and sample) algorithm [15], [47], [48].

Similar to feature-tracking algorithms, intensity based methods are categorized into two classes: gradient and correlation based algorithms.

The gradient scheme was first proposed in the field of engineering in order to calculate the speed of moving objects from a television signal [26], [54]. It estimates local motion by the computation of simultaneously measured spatial and temporal changes in the local light intensity of a moving image. This scheme in its mathematical form obtains an exact measurement of the local velocity  $\delta x/\delta t$  by dividing the temporal gradient  $\delta I/\delta t$  by the respective spatial gradient  $\delta I/\delta x$  of the pattern (x and t refer to the spatial variable and time, respectively;  $I$  denotes the light intensity) [11]. Hardware implementations of this algorithm have been demonstrated by Tanner et al. [78] and later by Deutschmann et al. [16].

Correlation based algorithms estimate the motion of an optical image by correlating intensity changes at different image regions. The computation is achieved by the interaction between the response of a visual sampling point and a delayed version of its neighboring sampling point response. The most popular correlation-based models are the Adelson-Bergen spatiotemporal energy model [2], the Hassenstein-Reichardt correlation model [31] and the Barlow-Levick motion detection model [6]. These models are utilized respectively to describe primate cortical complex cells [33], [60] to explain the optomotor response in flies [31] and direction selectivity in rabbit retina [6]. A variety of hardware implementations based on these algorithms have been realized. The Reichardt correlation model was implemented in hardware to estimate the velocity of image motion [4], [30], [66], [67]. In addition, Higgins and Korropati [37] implemented an analog VLSI sensor based on the Adelson-Bergen algorithm, and Benson and Delbruck [7] utilized null inhibition phenomena that are based on the Barlow-Levick model to implement an analog VLSI vision sensor. The Barlow-Levick model was realized in its correlation-based form and implemented in an analog VLSI vision sensor (R. Deutschmann, unpublished data, 2002).

## 1.2 Monolithic and multi-chip implementations

Motion detection algorithms can be realized in analog VLSI hardware implementations by using monolithic and multi-chip design strategies.

A monolithic motion sensor is defined as a single chip that has the photoreceptors and motion computation circuitry located on the focal plane. In this kind of implementation, integrating neuromorphic design principles with the capabilities of VLSI hardware systems allows designers to implement extremely efficient systems in terms of power and space. In addition, the fill factor of monolithic sensors can be maximized if the space used for each pixel is minimized. The fill factor is defined as the ratio of the photoreceptor area to the overall space used for motion computation in each pixel. Therefore, the main aim in this type of implementation is to decrease the computation performed in each pixel while maintaining a reliable and robust implementation of algorithms.

A monolithic implementation is composed of a 2-D array of pixels that perform motion computations and serial pixel scanners by which all the pixels can be read individually. By incorporating the scanners, real-time motion outputs can be read and raw data from photoreceptors can be easily obtained. This kind of implementation has been exploited to implement different motion computations in analog VLSI vision chips. For instance, heading direction and time-to-contact computations were realized successfully in monolithic vision chips by using this design strategy in building system level architectures [43].

In contrast to monolithic implementations, multi-chip sensors apply a modular strategy that helps to split the motion computation into different processing stages and realize them in multiple chips. In this study, we employed this strategy to realize intensity-based visual sensors that are composed of one photosensitive sender chip and one motion computing receiver chip. This kind of implementation is very efficient in increasing the computational capabilities of sensory systems and decreasing the computational overload in the front-end sensing stage. It employs an interchip communication protocol that is well-suited for achieving the communication between neuromorphic modules. This communication protocol was proposed by Mahowald [56] as a circuit analogy to the optic nerve and later Boahen [44] formalized and improved this protocol.

The multi-chip design strategy was successfully employed by Higgins et al. [36] to perform complex motion computations by incorporating a photosensitive sender chip which detects and transmits the position of moving spatial edges, and a receiver chip which performs 2D optical flow vector field computation by using the edge information. Boahen [9] used this strategy to implement binocular disparity-selective elements by interfacing two silicon retinas to three receiver chips. Moreover, Venier [81] implemented an orientation selective silicon retina with asynchronous interface. See [35] and [36] for a review of modular multi-chip neuromorphic architectures.

### 1.3 Obstacle avoidance

Visual information possesses different motion clues that assist organisms to employ parsimonious and minimalist strategies in dealing with different behavioral tasks. Recent studies on insect vision have elucidated the fact that insects can adapt to specific behavioral tasks by processing different spatiotemporal properties of visual information and capturing natural clues provided by their environment [23]. In this thesis, we focus on anatomical and physiological properties of flying insects in dealing with obstacle avoidance and develop a biologically inspired artificial system based on these properties.

Flying insects achieve obstacle avoidance and visual course control by employing mostly their visually guided behavioral mechanisms. Bees accomplish their visual course control by balancing the image velocity on their two eyes [74]. In contrast to bees, the anatomy of locusts is specialized in avoiding rapidly approaching objects [27] and therefore their largest motion detector is dedicated for object avoidance. In pigeons, similar responses have been obtained from nucleus rotundus [77]. This nucleus responds best to approaching objects that are on a collision course. It contains neurons that are sensitive to angular velocity and this information is used in time-to-collision computation [83].

These neural mechanisms that participate in collision avoidance systems can be classified into three groups: (1) the centering response which is mediated by a direction insensitive movement detecting system, (2) the turning response which prevents collision with obstacles such as walls and (3) the escape response which is initiated by the lobula giant movement detector in locusts to avoid predators. In this context, the second and third groups are studied under the same topic since they can be successfully incorporated to develop more robust artificial behaviors to avoid collisions.

### 1.4 Organization of the thesis

In this thesis, we present a variety of analog VLSI motion sensors that are implemented by exploiting monolithic and multi-chip design strategies. In addition, their algorithms and circuit level implementations are explained in detail. Lastly, a system implementation used for obstacle avoidance is described. The thesis is organized as follows.

In the second chapter, we explain a diversity of VLSI circuit blocks that have been employed in the design of motion sensors. In the third chapter, we describe two different monolithic visual motion sensors that are realized by hysteretic winner-take-all and nonlinear-differentiator based algorithms. In the fourth chapter, we present multi-chip implementations of the Adelson-Bergen spatiotemporal energy model, the Hassenstein-Reichardt correlation motion algorithm and the Barlow-Levick model. In the fifth chapter, we describe an obstacle avoidance algorithm based on the system level implementation of the Adelson-Bergen multi-chip sensor integrated with centering and escape behaviors. Finally, we discuss the advantages and disadvantages of the visual motion detector models, the obstacle avoidance system and their implementations.

## Chapter 2

# Analog VLSI building blocks

In this chapter, we explain a variety of analog VLSI circuit building blocks that we employed in our design of different neuromorphic architectures. Neuromorphic implementations are very suitable to realize biological models since transistors and neuron channels have similar characteristics [58]. Firstly, the distribution of carriers in transistors and neuron channels depend on the Boltzmann distribution. Secondly, ionic conductance is exponentially dependent on the voltage difference across the neuron membrane. Similarly, in the subthreshold operation mode a transistor has a current flow that is exponentially dependent on the voltages applied to its terminals.

High density and low power designs can be easily achieved with MOSFET transistors operating in the subthreshold region. Especially integrating this advantage with the capabilities of current mode design, which is conceived in building neuronal blocks, makes large and complex biomimetic architectures easy to realize. In this study, we explain the basics of MOSFET operation and its characteristics for both above-threshold and subthreshold regions. After that we describe the circuit details of the adaptive photoreceptor, transconductance amplifier, rectifier, current-comparator, absolute-value, squaring, nonlinear differentiator, winner-take-all and multiplier building blocks that we utilized to design a variety of motion detection sensors.

### 2.1 MOSFET operation

The metal-oxide-semiconductor field-effect transistor (MOSFET) is a four-terminal device with the terminals designated as gate, drain, source and substrate. The basic structure of a MOSFET is illustrated in Figure 2.1. An n-type MOSFET consists of a p-type silicon substrate and  $n+$  type drain and source regions formed in the substrate. The gate is usually realized by a metal or highly doped polysilicon and separated from the substrate by silicon dioxide [79]. In contrast to the n-type MOSFET, the p-type MOSFET has an n-well as a substrate and  $p+$  type drain and source regions.

In the operation of an n-type MOSFET, when there is no voltage present at its gate, the p-type silicon substrate is either in accumulation or in depletion and no current can flow between source and drain. Hence, the device acts like two back-to-back p-n junction diodes. In the second case when a reasonable amount of positive voltage is applied to the gate, the silicon surface of the transistor is inverted and a channel between source and drain is formed. In addition, if there is a voltage difference between the source and the drain terminals of the transistor a current flows through this channel.

The operation of an n-FET transistor can be divided into three regions, namely, cutoff, subthreshold (weak inversion) and above-threshold. In the subthreshold region current flows by diffusion while in the above-threshold regime current flows by drift.

The operation region of a MOSFET transistor is determined by its gate-to-source voltage,  $V_{GS}$ , relative to its predetermined threshold value. The threshold voltage,  $V_t$  is defined as the gate voltage when the surface potential or band bending reaches  $2\psi_B$  (where  $2\psi_B = (2kT/q)\ln(N_a/N_i)$ ) and the silicon charge is equal to the bulk depletion charge for that potential [79].

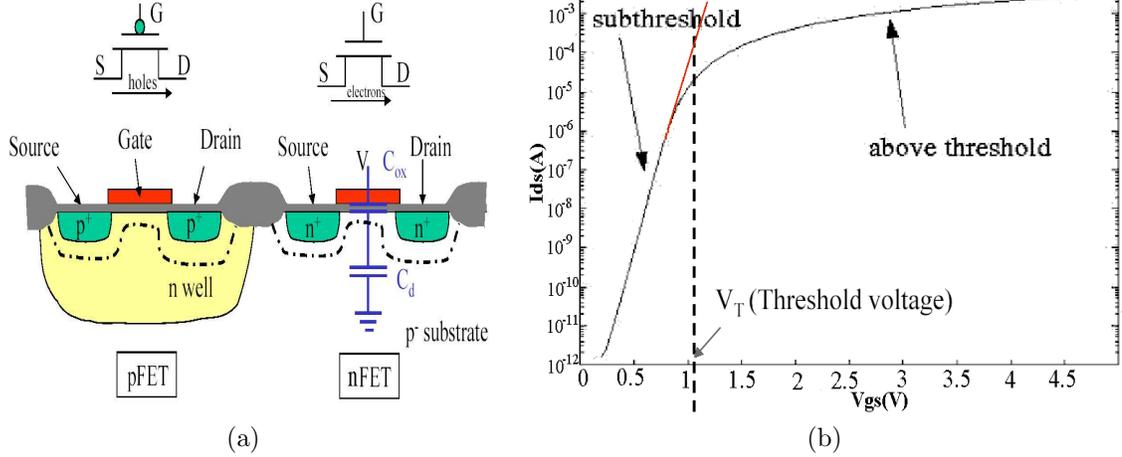


Figure 2.1: (a) Cross section of the MOSFET (b)  $I_{ds}$  versus  $V_{gs}$  of an n-type MOSFET, the threshold voltage ( $V_t$ ) is found at the voltage where the measured  $I_{ds}$  is half of the  $I_{ds}$  computed from the extrapolated exponential (Reproduced from [1] without permission).

If the gate-to-source voltage  $V_{GS}$  of an n-FET is bigger than the threshold voltage  $V_t$  then the transistor operates in the above-threshold region. This region itself has two operation modes depending on the value of the drain-to-source voltage,  $V_{DS}$ . For  $V_{DS} < V_{GS} - V_t$ , the transistor exhibits an ohmic characteristic and the relation between the drain-to-source current,  $I_{DS}$ , and the drain-to-source voltage  $V_{DS}$  for a constant  $V_{GS}$ , where  $V_{GS} > V_t$ , becomes as follows:

$$I_{DS} = \mu C_{ox} \frac{W}{L} [(V_{GS} - V_t)V_{DS} - \frac{V_{DS}^2}{2}] \quad (2.1)$$

where  $\mu$  is the mobility of electrons,  $C_{ox}$  is the oxide capacitance,  $W$  and  $L$  are the width and length, respectively, of the transistor. If  $V_{DS} > V_{GS} - V_t$ , the transistor saturates and the relation between the drain-to-source current,  $I_{DS}$ , and the gate-to-source voltage,  $V_{GS}$  for  $V_{GS} > V_t$  reduces to the following expression:

$$I_{DS} = \frac{\mu C_{ox}}{2} \frac{W}{L} (V_{GS} - V_t)^2 \quad (2.2)$$

In the second case, when  $V_{GS} < V_t$ , the n-FET operates in the subthreshold region. As is the case in the above threshold region, this operation mode also has two operation modes that determine the relation between  $I_{DS}$  and the transistor voltages. The general equation that holds for all these cases can be shown to be as follows:

$$I_{DS} = I_0 e^{\kappa V_G / V_T} (e^{-V_S / V_T} - e^{-V_D / V_T}) \quad (2.3)$$

where  $V_T$  is defined as the thermal voltage and is equal to  $kT/q = 25mV$  at room temperature, and  $\kappa$  is the back gate coefficient. In addition,  $I_0$  is a transistor parameter and  $I_0 = \frac{W}{L} q D N_\phi e^{-\frac{\phi_0}{kT/q}}$ . For  $V_{DS} > 4kT/q$ , the transistor saturates and the drain-to-source current  $I_{DS}$  becomes,

$$I_{DS} = I_0 e^{(\kappa V_G - V_S) / V_T} \quad (2.4)$$

In the following sections, we describe a variety of circuit blocks that utilize the explained characteristics of the MOSFET.



output node ( $v_{prouit}$ ), the transfer function,  $v_{prouit}/i_{in}$ , of the photoreceptor circuit can be found by using small-signal approximation.

$$\frac{v_{prouit}}{i_{in}} = \frac{1}{g_{m1}} \left[ \frac{\frac{s(\tau_{ld} + \tau_l) + g_a/g_{m3}}{s\tau_l + g_a/g_{m3}}}{(s\tau_r + 1/\kappa)(1/A_{amp} + s\tau_{ld} + \frac{s\tau_{ld}}{A_{amp}(s\tau_l + g_a/g_{m3})}) + 1} \right] \quad (2.5)$$

where  $g_a$  is the output conductance of transistor  $M_2$ ,  $C_r$  is the parasitic capacitance at the gate of  $M_3$  and  $A_{amp} = \frac{g_{m3}}{g_d}$ . In this thesis, we represent the transconductance as  $g_m$  and the output conductance of the circuit as  $g_d$ . Hence, the time constants are defined as follows:

$$\tau_l = \frac{C_l}{g_{m3}}; \tau_r = \frac{C_r}{g_{m1}}; \tau_{ld} = \frac{C_d}{g_{m5}} \quad (2.6)$$

The transfer function of the circuit verifies the experimental results shown in Figure 2.2c. The photoreceptor exhibits a band-pass characteristic for high intensity levels and shows a low-pass characteristic for lower intensity levels in the frequency range from 1 to 100 Hz. Its cut-off frequency depends on the intensity level of the input signal and its gain for flat regions is set by the capacitive ratio  $A_C = \frac{C_1 + C_2}{C_1}$ .

For short time scales, the adaptive photoreceptor behaves like a high gain inverting amplifier of  $V_{S_{M1}}$  with a gain  $A = g_m(\frac{r_{on}r_{op}}{r_{on} + r_{op}})$  where  $r_{on}$  is the output resistance of transistor  $M_3$  and  $r_{op}$  is the output resistance of transistor  $M_4$ . For long time scales,  $V_{prouit}$  adapts to  $V_{fb}$  and therefore its output is determined by its adaptation state. The characteristics of the adaptive photoreceptor can be summarized as follows:

Adapted signal:

$$V_{fb} = \kappa^{-1} [V_{GS_{M3}} + V_T \ln(\frac{I_{DS_{M1}}}{I_0})] \quad (2.7)$$

Transient amplification:

$$A_C = \frac{C_1 + C_2}{C_1} \quad (2.8)$$

Transient signal:

$$V_{prouit} = A_C V_{fb} \quad (2.9)$$

Small signal response:

$$dV_{prouit} = A_C V_T \frac{A}{\kappa A - 1} \frac{dI_{DS_{M1}}}{I_{DS_{M1}}} \approx A_C \frac{V_T}{\kappa} \frac{dI_{DS_{M1}}}{I_{DS_{M1}}} \quad (2.10)$$

where A is the amplification gain of the inverting amplifier. If we neglect the DC offsets of  $V_{GS_{M3}}$  and  $V_{prouit}$ , then  $V_{prouit}$  and  $V_{fb}$  can be approximated as follows:

$$V'_{prouit} = \frac{V_T}{\kappa} A_C \ln(\frac{I_{DS_{M1}}}{I_0}) \quad (2.11)$$

$$V'_{fb} = \frac{V_T}{\kappa} \ln(\frac{I_{DS_{M1}}}{I_0}) \quad (2.12)$$

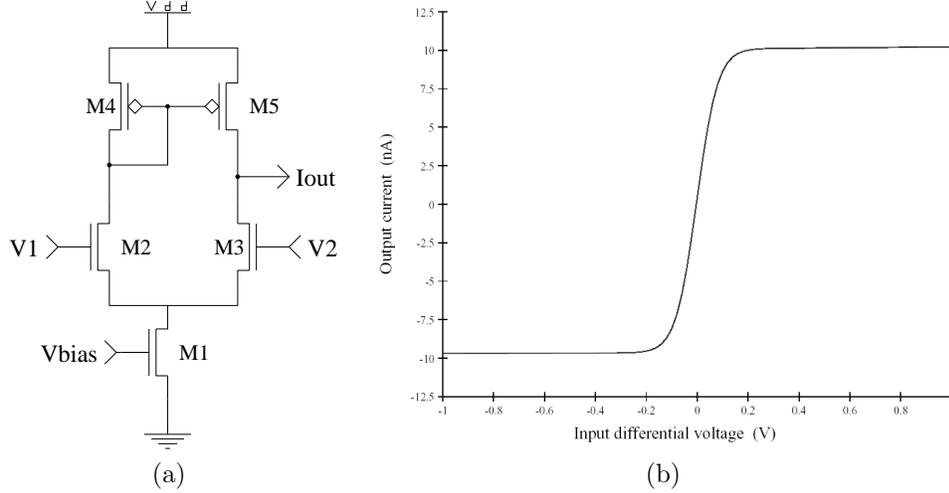


Figure 2.3: (a) Transconductance amplifier circuit. (b) Current response to a range of differential input voltages. Reproduced from [42] without permission.

## 2.3 Differential transconductance amplifier

The transconductance amplifier circuit illustrated in Figure 2.3a is used to convert differential input voltage signals to a current output. It provides a saturated nonlinear current output. This circuit can also be used as a differential voltage amplifier if instead of the current output  $I_{out}$ , the drain voltage of transistor  $M_3$  is taken as an output.

The transconductance amplifier can operate in the above-threshold and subthreshold regions. In subthreshold, the current response of the circuit to a differential input voltage can be found by analyzing the circuit using subthreshold equations (see Section 2.1). Considering the voltages  $V_1$  and  $V_2$  illustrated in Figure 2.3a, the current through transistors  $M_1$ ,  $M_2$  and  $M_3$  can be found as follows:

$$I_{DS_{M_2}} = I_0 e^{\frac{\kappa V_1 - V_{DS_{M_1}}}{V_T}} \quad (2.13)$$

$$I_{DS_{M_3}} = I_0 e^{\frac{\kappa V_2 - V_{DS_{M_1}}}{V_T}} \quad (2.14)$$

$$I_{DS_{M_1}} = I_{DS_{M_2}} + I_{DS_{M_3}} = I_0 e^{\frac{\kappa V_{bias}}{V_T}} \quad (2.15)$$

where  $V_{DS_{M_1}}$  is the drain-to-source voltage of transistor  $M_1$  and  $V_{bias}$  is the gate-to-source voltage of transistor  $M_1$ . The current through transistor  $M_1$  can be derived in terms of input voltages  $V_1$  and  $V_2$  as in the following:

$$I_{DS_{M_1}} = I_0 e^{\frac{-V_{DS_{M_1}}}{V_T}} \left( e^{\frac{\kappa V_1}{V_T}} + e^{\frac{\kappa V_2}{V_T}} \right) \quad (2.16)$$

From this equation  $V_{DS_{M_1}}$  can be found and this helps to define the currents  $I_{DS_{M_1}}$  and  $I_{DS_{M_2}}$  in terms of the input voltages  $V_1$  and  $V_2$  in the following way:

$$e^{\frac{-V_{DS_{M_1}}}{V_T}} = \frac{I_{DS_{M_1}}}{I_0} \frac{1}{e^{\frac{\kappa V_1}{V_T}} + e^{\frac{\kappa V_2}{V_T}}} \quad (2.17)$$

$$I_{DS_{M_2}} = I_{DS_{M_1}} \frac{e^{\frac{\kappa V_1}{V_T}}}{e^{\frac{\kappa V_1}{V_T}} + e^{\frac{\kappa V_2}{V_T}}} \quad (2.18)$$

$$I_{DS_{M_3}} = I_{DS_{M_1}} \frac{e^{\frac{\kappa V_2}{V_T}}}{e^{\frac{\kappa V_1}{V_T}} + e^{\frac{\kappa V_2}{V_T}}} \quad (2.19)$$

The transconductance amplifier shown in Figure 2.3a obtains its output current by subtracting the currents  $I_{DS_{M_2}}$  and  $I_{DS_{M_3}}$  as follows:

$$I_{out} = I_{DS_{M_2}} - I_{DS_{M_3}} = I_{DS_{M_1}} \frac{e^{\frac{\kappa V_1}{V_T}} - e^{\frac{\kappa V_2}{V_T}}}{e^{\frac{\kappa V_1}{V_T}} + e^{\frac{\kappa V_2}{V_T}}} \quad (2.20)$$

$$I_{out} = I_{DS_{M_1}} \tanh\left(\frac{\kappa}{2V_T}(V_1 - V_2)\right) \quad (2.21)$$

For small voltage differences the circuit yields a linear relationship between the differential input voltage ( $V_1 - V_2$ ) and the current output  $I_{out}$ , but for larger voltage differences the output current of the transconductance amplifier saturates as depicted in Figure 2.3b. For  $|V_1 - V_2| < 200mV$ ,  $I_{out}$  can be reduced to an expression,

$$I_{out} \approx g_m(V_1 - V_2) \quad (2.22)$$

The transconductance ( $g_m$ ) and output conductance ( $g_d$ ) of the amplifier is

$$g_m = \frac{I_{DS_{M_1}} \kappa}{2V_T} \quad (2.23)$$

$$g_d = -\frac{\delta I_{out}}{\delta V_{out}} \approx \frac{I_{DS_{M_1}}}{V_E} \quad (2.24)$$

where  $V_E$  is the Early voltage of transistors  $M_3$  and  $M_5$ .

When the transconductance amplifier is used as a differential voltage amplifier, the voltage output, which is the drain voltage of  $M_3$ , can be found by considering the conditions applied by the differential input voltage. Assuming that transistors  $M_2$  and  $M_4$  are always in saturation, then the voltage limitations can be expressed as follows:

- To keep  $M_5$  is in saturation

$$V_{out} < V_{DD} - 4V_T \quad (2.25)$$

- To keep  $M_3$  in saturation

$$V_{out} > \kappa(\max(V_1, V_2) - V_{bias}) + 4V_T \quad (2.26)$$

The saturation conditions of transistors  $M_3$  and  $M_5$  restrict the output voltage range.

In the above threshold region, the current output of the transconductance amplifier can be found as follows:

$$I_{out} = \frac{\beta}{2}(V_1 - V_2) \sqrt{\frac{4I_{DS_{M_1}}}{\beta} - (V_1 - V_2)^2} \quad (2.27)$$

where  $\beta = \mu C_{ox} \frac{W}{L}$ . In this operation mode, for  $|V_1 - V_2| < \sqrt{\frac{2I_{DSM_1}}{\beta}}$ , the transconductance becomes

$$g_m = \sqrt{\beta I_{DSM_1}} \quad (2.28)$$

When this circuit is used as a differential voltage amplifier in the above threshold operation mode, the voltage output and amplification gain can be both found as follows:

$$V_{out} = A(V_1 - V_2) \quad (2.29)$$

$$A = \frac{dV_{out}}{d(V_1 - V_2)} = \frac{g_m}{g_d} \approx \sqrt{\frac{\beta}{I_{DSM_1}}} V_E \quad (2.30)$$

where  $V_E$  is the Early voltage. As can be observed from the formula, the open-circuit voltage gain,  $A$ , increases with the Early voltage.

## 2.4 Full-wave rectifier and current comparator

A full wave rectifier is a current mode circuit used to separate a bidirectional input current into two unidirectional currents that represent the negative and positive parts of the input current. The full-wave rectifier circuit is illustrated in Figure 2.4a. The output currents of this circuit can be formulated in terms of the input current  $I_{in}$  as follows:

$$I_{positive} = \begin{cases} I_{in} & \text{if } I_{in} > 0 \\ 0 & \text{if } I_{in} < 0 \end{cases}$$

$$I_{negative} = \begin{cases} 0 & \text{if } I_{in} > 0 \\ |I_{in}| & \text{if } I_{in} < 0 \end{cases}$$

where  $I_{negative}$  flows into  $M_1$  and  $M_4$  and  $I_{positive}$  flows out of  $M_2$  and  $M_3$ . This circuit is used for bidirectional input currents that provide or extract current to/from the circuit. If an input current flows into the rectifier then transistor  $M_1$  opens its path and this current flows into transistor  $M_4$ . In this case the source-to-gate voltage  $V_{SG}$  of transistor  $M_1$  increases to supply enough current. This causes transistor  $M_2$  to close because the gate-to-source voltage  $V_{GS}$  of this transistor goes to some value smaller than zero relative to  $V_{fwrbias}$ . In the second case, when the input current flows out of the circuit, the upper path provided by transistors  $M_2$  and  $M_3$  opens and the other path formed by transistors  $M_1$  and  $M_4$  closes. The most important transistors in the operation of the circuit are  $M_1$  and  $M_2$ , because they compare the voltage effect of the input current with the bias voltage  $V_{fwrbias}$  and determine which path to open. Furthermore, this circuit works quite efficiently because the  $V_{GS}$  of transistor  $M_2$  and the  $V_{SG}$  of transistor  $M_1$  can not be positive at the same time and therefore only one path becomes open for each case.

In Figure 2.4b, the current comparator circuit is shown. This circuit is used to compare the positive and negative parts of a bidirectional input current. The bidirectional current is separated into positive and negative currents by employing a full-wave rectifier circuit, and the voltage input-output characteristic of the overall circuit is improved by using two inverter circuits. For the reason that the full-wave circuit is a conditional circuit, the negative and positive currents cannot both be bigger than zero at the same time, i.e., when one of the currents is bigger than zero, the other one has to be zero. These currents are mirrored and then compared. When one of the currents becomes bigger than the other one, the drain-to-source voltage  $V_{DS}$  of the transistor carrying the bigger current goes down in magnitude to limit or shut down the current from that transistor. Hence it can be concluded that the existence of a negative or positive current determines the voltage output at the input of the first inverter. The final output will be  $V_{DD}$  if the input current flows out of the circuit, and will be 0 if the current flows into the circuit.

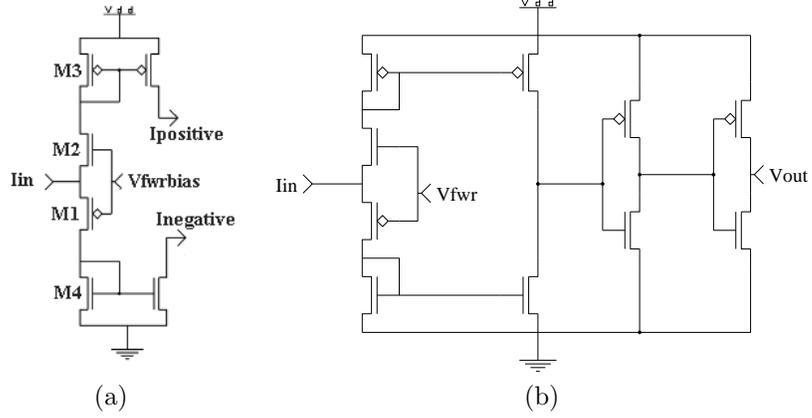


Figure 2.4: (a) The full wave rectifier circuit. (b) The current comparator circuit.

## 2.5 Absolute-value and squaring circuits

The absolute value circuit shown in Figure 2.5a was inspired by an above-threshold circuit of Bult and Wallinga [12]. This circuit takes a bidirectional current as an input and performs the computation by operating conditionally.

In the case when an input current flows out of the circuit, the drain-to-source voltage  $V_{DS}$  of diode-connected transistor  $M_2$  increases while the source-to-gate voltage  $V_{SG}$  of transistor  $M_1$  decreases relative to the gate voltage  $V_{abs}$  and eventually retards any current flowing into it. In the other case, when current flows into the circuit, the  $V_{GS}$  of transistor  $M_2$  decreases to close that transistor, and at the same time the  $V_{SG}$  of transistor  $M_1$  increases and opens the path for the current to flow. In this way it provides a conditional path for the bidirectional current to flow out of the circuit. In both cases, the output current is obtained as directed out of the circuit. The final output can be represented as follows:

$$I_{rect} = |I_{in}| \quad (2.31)$$

The squaring circuit is a translinear circuit that computes the square of input current signals [80]. Its circuit schematic is shown in Figure 2.5b. As illustrated in the figure, the input current flows into node  $N_3$  and the squared current  $I_{sq}$  flows into node  $N_5$ . The relation between the input and output currents can be obtained by utilizing the subthreshold I-V characteristics of MOSFETs and translinear principles [80].

Let the voltage at node  $N_3$  be  $V_a$  and the voltage at node  $N_4$  be  $V_b$ . If we neglect the Early effect, the currents through transistors  $M_4$ ,  $M_5$  and  $M_6$  can be formulated as follows:

$$I_4 = I_0 e^{\frac{\kappa(V_a - V_b)}{V_T}} \quad (2.32)$$

$$I_5 = I_0 e^{\frac{\kappa(V_b)}{V_T}} \quad (2.33)$$

$$I_6 = I_0 e^{\frac{\kappa(V_a)}{V_T}} \quad (2.34)$$

where  $V_T$  is defined as the thermal voltage and equal to  $kT/q = 25mV$  at room temperature, and  $\kappa$  is the back gate coefficient. By arranging the above terms and using the equalities  $I_4 = I_5 = I_{in}$  and  $I_6 = I_{sq}$ , we can obtain the following expressions:

$$I_{in} = I_0 e^{\frac{\kappa(V_a)}{V_T}} e^{\frac{\kappa(-V_b)}{V_T}} \quad (2.35)$$

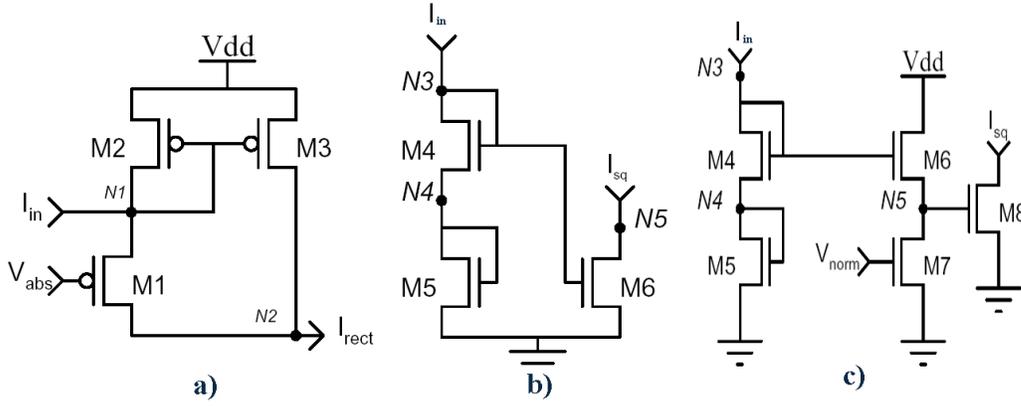


Figure 2.5: The full wave rectifier and normalized squaring circuit. (a) The absolute value circuit used to rectify the incoming bidirectional current  $I_{in}$ . The output current  $I_{rect}$  is rectified and flows out of the node  $N_2$ . (b) The squaring circuit. It receives the rectified current  $I_{in}$  as the input and the output is a squared current  $I_{sq}$  at the node  $N_5$ . (c) The normalized squaring circuit. The current level in the circuit can be adjusted using the bias voltage  $V_{norm}$ .

$$I_{in} = I_0 e^{\frac{\kappa(V_b)}{V_T}} \quad (2.36)$$

$$I_{sq} = I_0 e^{\frac{\kappa(V_a)}{V_T}} \quad (2.37)$$

By combining the above equations we can find the following final expression:

$$I_{sq} = \frac{I_{in}^2}{I_0} \quad (2.38)$$

The circuit performs a squaring operation and scales the output current with  $I_0$ . However, we do not have an explicit control on the magnitude of the output current,  $I_{sq}$ . If the current level is high after squaring, then the transistor  $M_6$  may operate in above threshold and the above formula will not hold. In order to prevent this case from happening, an alternative squaring circuit illustrated in Figure 2.5c can be employed.

The normalized squaring circuit shown in Figure 2.5c yields a relation between the input current  $I_{in}$  and the squared output current  $I_{sq}$  in terms of the normalizing current  $I_{norm}$ .

Let the voltages at node  $N_3$ ,  $N_4$  and  $N_5$  be  $V_a$ ,  $V_b$  and  $V_c$  respectively. The expressions for the currents through transistors  $M_4$ ,  $M_5$ ,  $M_6$ ,  $M_7$  and  $M_8$  become as follows:

$$I_4 = I_0 e^{\frac{\kappa(V_a - V_b)}{V_T}} \quad (2.39)$$

$$I_5 = I_0 e^{\frac{\kappa(V_b)}{V_T}} \quad (2.40)$$

$$I_6 = I_0 e^{\frac{\kappa(V_a - V_c)}{V_T}} \quad (2.41)$$

$$I_7 = I_{norm} = I_0 e^{\frac{\kappa(V_{norm})}{V_T}} \quad (2.42)$$

$$I_8 = I_0 e^{\frac{\kappa(V_c)}{V_T}} \quad (2.43)$$

Since  $I_4 = I_5 = I_{in}$ ,  $I_6 = I_7$  and  $I_8 = I_{sq}$ ,

$$V_c = \frac{V_T}{\kappa} \ln \frac{I_{sq}}{I_0} \quad (2.44)$$

$$V_a = 2V_b \quad (2.45)$$

$$I_{sq} = \frac{I_{in}^2}{I_{norm}} \quad (2.46)$$

By utilizing the normalized squaring circuit we can have control over the squared current  $I_{sq}$  and limit the power consumption.

## 2.6 Nonlinear differentiator

The nonlinear differentiator circuit shown in Figure 2.6a operates by utilizing the adaptive nonlinear filter characteristic of a diode-connected transistor connected in parallel with a capacitor. The most crucial property of this circuit is that it provides a decaying pulse response with a small time constant when its output voltage is large and a large time constant when its output voltage is small.

In Figure 2.6c, a diode-connected transistor in parallel with a capacitor is illustrated. The capacitor current  $I_c$  depends on capacitor voltage  $V_{out}$ ,

$$I_c(t) = C \frac{\delta V_{out}(t)}{\delta t} \quad (2.47)$$

In the subthreshold region, the transistor has an I-V characteristic that can be formulated as follows:

$$I_m(t) = I_0 e^{\frac{\kappa V_{out}(t)}{V_T}} \quad (2.48)$$

By combining Equations 2.47 and 2.48, the currents in this circuit can be written as,

$$I_{in}(t) = I_m(t) + I_c(t) \quad (2.49)$$

$$I_{in}(t) = I_0 e^{\frac{\kappa V_{out}(t)}{V_T}} + C \frac{\delta V_{out}(t)}{\delta t} \quad (2.50)$$

The initial condition of the circuit at  $t = 0$ ,

$$I_{in}(0) = I_0 e^{\frac{\kappa V_{out}(t=0)}{V_T}} + C \frac{\delta V_{out}(t)}{\delta t} \Big|_{t=0} \quad (2.51)$$

When the input current  $I_{in}$  is switched off, the capacitor discharges through the diode connected transistor. For  $t > 0$ ,

$$I_m(t) = -I_c(t) \quad (2.52)$$

$$I_0 e^{\frac{\kappa V_{out}(t)}{V_T}} = -C \frac{\delta V_{out}(t)}{\delta t} \quad (2.53)$$

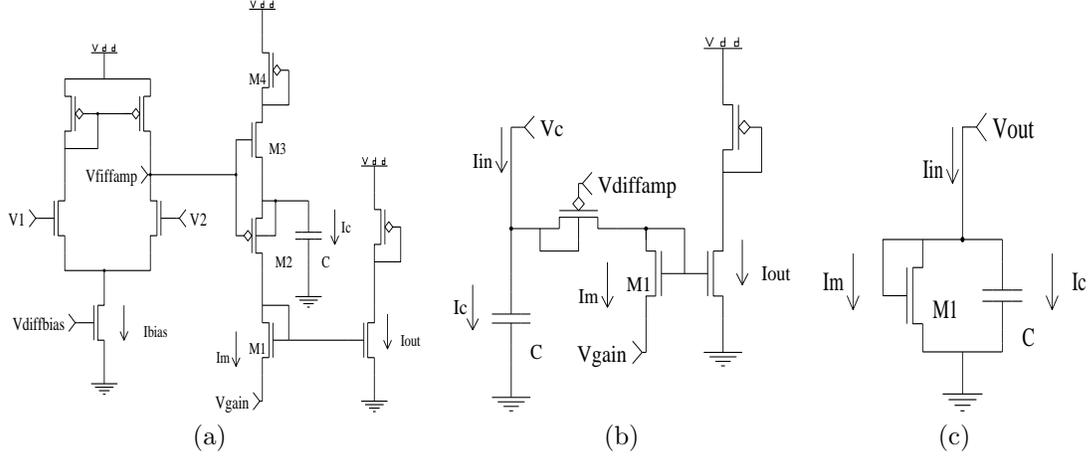


Figure 2.6: The nonlinear differentiator circuit and its equivalent circuits are illustrated for different conditions. (a) The nonlinear differentiator circuit is shown. In this circuit  $V_1$  and  $V_2$  represent the input voltages and  $I_{out}$  is the output current of the circuit. (b) The equivalent circuit of the nonlinear differentiator circuit that is used to show the relation between  $I_{in}$  and  $I_{out}$ . (c) The adaptive nonlinear filter formed by a diode-connected transistor connected in parallel with a capacitor. This circuit is used to analyze the nonlinear differentiator circuit when  $M_2$  is shorted.

$$-\frac{I_0}{C}\delta t = e^{-\frac{\kappa V_{out}}{V_T}} \delta V_{out} \quad (2.54)$$

Integrating both sides,

$$-\frac{I_0}{C}t = -\frac{V_T}{\kappa} e^{-\frac{\kappa V_{out}}{V_T}} \Big|_{V_{out}=V_{out}(t=0)}^{V_{out}=V_{out}(t)} \quad (2.55)$$

$$\frac{\kappa I_0}{CV_T} t = e^{-\frac{\kappa V_{out}(t)}{V_T}} - e^{-\frac{\kappa V_{out}(t=0)}{V_T}} \quad (2.56)$$

$$V_{out} = \frac{V_T}{\kappa} \ln \left( \frac{e^{\frac{\kappa V_{out}(t=0)}{V_T}}}{1 + \frac{\kappa I_0 e^{-\frac{\kappa V_{out}(t=0)}{V_T}}}{CV_T} t} \right) \quad (2.57)$$

By using Equation 2.48, for  $t > 0$ ,  $I_m(t)$  can be expressed as follows:

$$I_m(t) = \frac{I_0 e^{\frac{\kappa V_{out}(t=0)}{V_T}}}{1 + \frac{\kappa I_0 e^{-\frac{\kappa V_{out}(t=0)}{V_T}}}{CV_T} t} \quad (2.58)$$

For  $t \gg \frac{CV_T}{\kappa I_0} e^{-\frac{\kappa V_{out}(t=0)}{V_T}}$ , this circuit becomes independent of the initial output voltage, and the output current and voltage become

$$I_m(t) = \frac{CV_T}{\kappa t} \quad (2.59)$$

$$V_{out}(t) = \frac{V_T}{\kappa} \ln \left( \frac{I_m(t)}{I_0} \right) = \frac{V_T}{\kappa} \ln \left( \frac{CV_T}{\kappa I_0 t} \right) \quad (2.60)$$

In this study we integrate this property of the adaptive nonlinear circuit with a differential amplifier to build a differentiator circuit. This circuit is illustrated in Figure 2.6a and utilized to take the derivative of the amplified difference of two input signals. It extracts the negative edge of the differential amplifier output. At the crossing point of the input signals it yields a decaying pulse.

The differential amplifier sets the conditional responses and timing of the circuit. When  $V_1$  is bigger than  $V_2$ , the differential voltage output will go high (close to  $V_{dd}$ ). In this time interval, since the voltage on capacitor  $C$  will be less than the differential amplifier voltage output, transistor  $M_2$  will get closed and the current output  $I_m$  of the circuit diminishes. On the other hand, transistors  $M_3$  and  $M_4$  open their path for a small time interval to charge the capacitor. The current output of these transistors flows into the capacitor and charges it to a certain voltage. In the second case when  $V_1$  is smaller than  $V_2$ , the differential voltage output goes low (close to zero). This will cause transistors  $M_3$  and  $M_4$  to close their path and transistors  $M_6$  and  $M_7$  to open their path to let the current stored in the capacitor to flow through these transistors. This case is illustrated in Figure 2.6c. In this circuit, the upper path provided by transistors  $M_3$  and  $M_4$  is closed and therefore can be ignored in the analysis. In addition,  $I_C$  becomes negative since  $V_C$  is bigger than  $V_{diffamp}$  and  $I_{in}$  is zero. The charge stored in the capacitor flows into  $M_1$  and the currents through capacitor and transistor  $M_1$  are related as  $I_m = -I_C$ . This circuit configuration can be related to the case where a diode-connected transistor is in parallel with a capacitor (Figure 2.6c). If we ignore the effect of pass-transistor  $M_2$ , the capacitor and transistor  $M_1$  form the same configuration except this time the source voltage of transistor  $M_1$  is  $V_{gain}$ . For  $t > 0$ , the output current of the circuit  $I_{out}$  becomes

$$I_m(t) = \frac{I_0 e^{\frac{\kappa V_C(t=0)}{V_T}} e^{\frac{V_{gain}}{V_T}}}{1 + \frac{\kappa I_0 e^{\frac{\kappa V_C(t=0)}{V_T}}}{C V_T} t} \quad (2.61)$$

For  $t \gg \frac{C V_T}{\kappa I_0} e^{-\frac{\kappa V_C(t=0)}{V_T}}$  (when the circuit becomes independent of the initial voltage  $V_C(t=0)$ ), the current output  $I_{out}$  can be formulated as follows:

$$I_{out}(t) = \frac{C V_T}{\kappa t} e^{\frac{V_{gain}}{V_T}} \quad (2.62)$$

The simulated response of the circuit  $I_{out}$  is illustrated in Figure 2.7 for the case where a 10Hz sinusoidal input signal is introduced as a stimulus at  $V_1$  while the second input,  $V_2$ , is held constant.

## 2.7 Winner-take-all circuit

The winner-take-all circuit shown in Figure 2.8a is a current mode circuit that was inspired by the inhibitory mechanisms present in the nervous system. This circuit was originally designed and presented by Lazzaro [49]. It is advantageous in terms of power consumption and silicon area usage since it processes input signals by using very few transistors and interconnections. Hence it is preferred in a wide variety of applications that utilize nonlinear inhibition mechanism [17], [38], and [76].

The hysteretic winner-take-all circuit (Figure 2.8b) is a task specific version of the winner-take-all circuit and was proposed by Deweerth et al [18]. It allows the winning input to maintain its status without having to reset the network unless another input exceeds the sum of the winning and bias inputs.

In this study, we employ a two-input winner-take-all circuit for motion computation. Therefore, we explain the working principles of the circuits only for the two-input case. The simple winner-take-all circuit (Figure 2.8a) compares two inputs and encodes the logarithm of the winner input. The output of the circuit depends on the relative amplitude of the input currents.

Firstly, assuming that input currents  $I_{in1}$  and  $I_{in2}$  created by  $V_{in1}$  and  $V_{in2}$  are equal, transistors  $M_3$  and  $M_7$  drive the same amount of current and by symmetry have same voltages  $V_{SD}$  and  $V_{SG}$ .

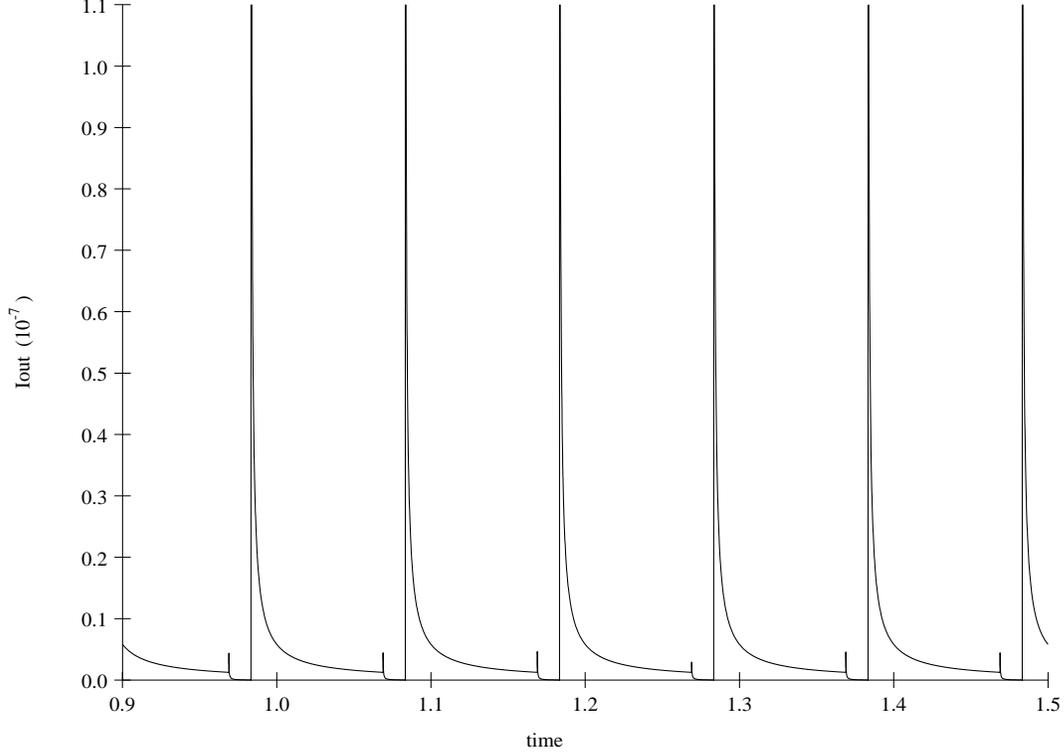


Figure 2.7: Current output of the nonlinear differentiator circuit to a 10Hz stimulus.

This will result in having the same voltage outputs at  $V_{wta1}$  and  $V_{wta2}$ . Therefore  $M_2$  and  $M_6$  share the current  $I_b$ , which is set by the bias  $V_b$ .

$$I_{M_2} = I_{M_7} = \frac{I_b}{2} \quad (2.63)$$

In the subthreshold region, if the drain-to-source voltage of transistor  $M_4$  is defined as  $V_{DS4}$ , then the currents flowing into transistors  $M_3$  and  $M_7$  can be shown to be as follows:

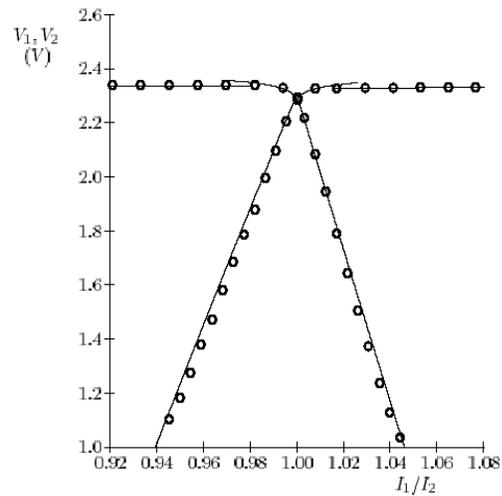
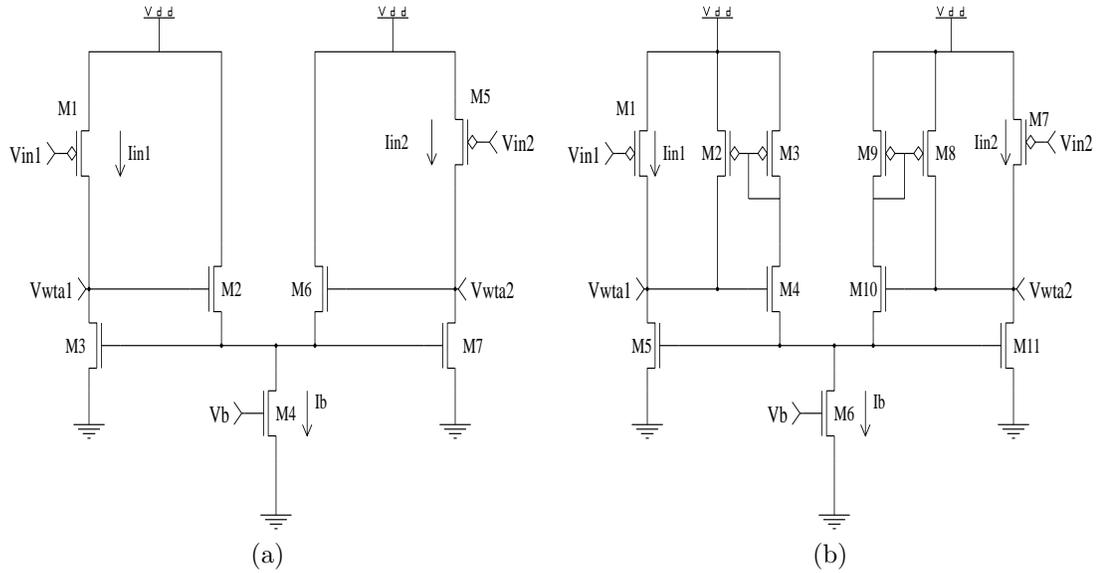
$$I_{M_3} = I_{M_7} = I_0 e^{\frac{\kappa V_{DS4}}{V_T}} \quad (2.64)$$

where  $I_0$  is the fabrication parameter and  $V_T = kT/q$ . The voltage outputs at  $V_{wta1}$  and  $V_{wta2}$  can be formulated as follows:

$$V_{wta1} = \frac{V_T}{\kappa} \ln\left(\frac{I_{in1}}{I_0}\right) + \frac{V_T}{\kappa} \ln\left(\frac{I_b}{2I_0}\right) \quad (2.65)$$

$$V_{wta2} = \frac{V_T}{\kappa} \ln\left(\frac{I_{in2}}{I_0}\right) + \frac{V_T}{\kappa} \ln\left(\frac{I_b}{2I_0}\right) \quad (2.66)$$

In the second case, when  $I_{in1}$  is larger than  $I_{in2}$ , the equality of currents does not hold. At first, any change in  $I_{in1}$  causes the drain voltage of  $M_3$  to increase and since this voltage is also the gate voltage of the transistor  $M_2$ , its source voltage must go up to keep its current the same. In turn this helps  $M_3$  drive  $\delta I$  more current. However, transistors  $M_3$  and  $M_7$  share the same gate voltage and therefore any increase in this voltage has the same effect on the other side of the circuit. If the gate voltage of transistor  $M_7$  increases while the input current is still  $I_{in2}$ , then  $V_{wta2}$  decreases to compensate, i.e., suppress the extra current. The Early effect helps the voltage  $V_{wta2}$  decrease



(c)

Figure 2.8: Schematics of the winner-take-all and hysteretic winner-take-all circuits and experimental data for a two-input winner-take-all circuit. (a) The current mode two-input winner-take-all circuit. (b) The current mode two-input hysteretic winner-take-all circuit. (c) Experimental data (circles) and theoretical statements (solid lines) for a two-input winner-take-all circuit. Reproduced without permission from [49].

linearly for small  $\delta I$  currents; but for large  $\delta I$  values the drain voltage of  $M_7$  must decrease down towards zero to force the transistor to leave saturation. For large  $\delta I$  values, the final result of the winner output  $V_{wta1}$  and the loser output  $V_{wta2}$  can be shown to be as follows:

$$V_{wta1} = \frac{V_T}{\kappa} \ln\left(\frac{I_{in1}}{I_0}\right) + \frac{V_T}{\kappa} \ln\left(\frac{I_b}{I_0}\right) \quad (2.67)$$

$$V_{wta2} \approx 0 \quad (2.68)$$

If the input current  $I_m$  and the output voltage  $V_m$  are defined as the current and voltage levels when the input currents  $I_{in1}$  and  $I_{in2}$  are equal, then these voltages and currents can be written as  $V_{wta1} = V_{wta2} = V_m$  and  $I_{in1} = I_{in2} = I_m$ . By using these, for small  $\delta I$  differences between  $I_{in1}$  and  $I_{in2}$ , if  $V_{wta1}$  is the winner and  $V_{wta2}$  is the loser, the voltage outputs that determine the linear range of the circuit can be formulated as follows:

$$V_{wta1} = V_T \ln \frac{I_{in1}}{I_m} + V_T \ln \frac{V_E}{V_T} + V_m \quad (2.69)$$

$$V_{wta2} = \frac{V_T}{2} + V_m - V_E \ln \frac{I_{in1}}{I_m} \quad (2.70)$$

As a result, the winner output will encode the logarithm of the associated input. The relation between the output voltages  $V_{wta1}$  and  $V_{wta2}$  for different ratios of input currents is illustrated in Figure 2.8c.

In contrast to the winner-take-all circuit, the hysteretic winner-take-all circuit (Figure 2.8b) awards the bias current to the winner through transistors  $M_2$  and  $M_3$  or  $M_8$  and  $M_9$  and therefore the loser input has to exceed the sum of the winner input and bias current to be able to become the new winner. The race between the input currents happens in the same way as in the winner-take-all circuit.

## 2.8 One and four-quadrant multipliers

In this section one and four-quadrant current mode multipliers are described. These circuits operate in the subthreshold region and therefore consume little power. They are employed to obtain nonlinearities that are needed to implement excitatory and inhibitory connections in neuromorphic applications.

The one-quadrant multiplier shown in Figure 2.9a can be analyzed in the subthreshold region by using the translinear principle for the loop formed by transistors  $M_3$ ,  $M_5$ ,  $M_7$  and  $M_8$  [84]. The currents of these transistors can be related as follows:

$$I_{M3}I_{M5} = I_{M7}I_{M8} \quad (2.71)$$

where  $I_{M3}$  is the current  $I_{in}$  set by  $V_{in}$ .  $I_{M5}$  is set by  $V_w$  and  $I_{M7}$  is the bias set by  $V_b$  and therefore  $I_{M5} = I_w$  and  $I_{M7} = I_b$ . The relation between these currents determine the value of the current output.

$$I_{out} = \frac{I_{in}I_w}{I_b} \quad (2.72)$$

The four-quadrant current multiplier illustrated in Figure 2.9b is used to multiply two bidirectional input currents [13]. This circuit operates under the condition that  $|I_{in1}|, |I_{in2}| < I_b$  where input currents can be positive or negative. As shown and formulated in [13],  $I_{R1}$  can be represented

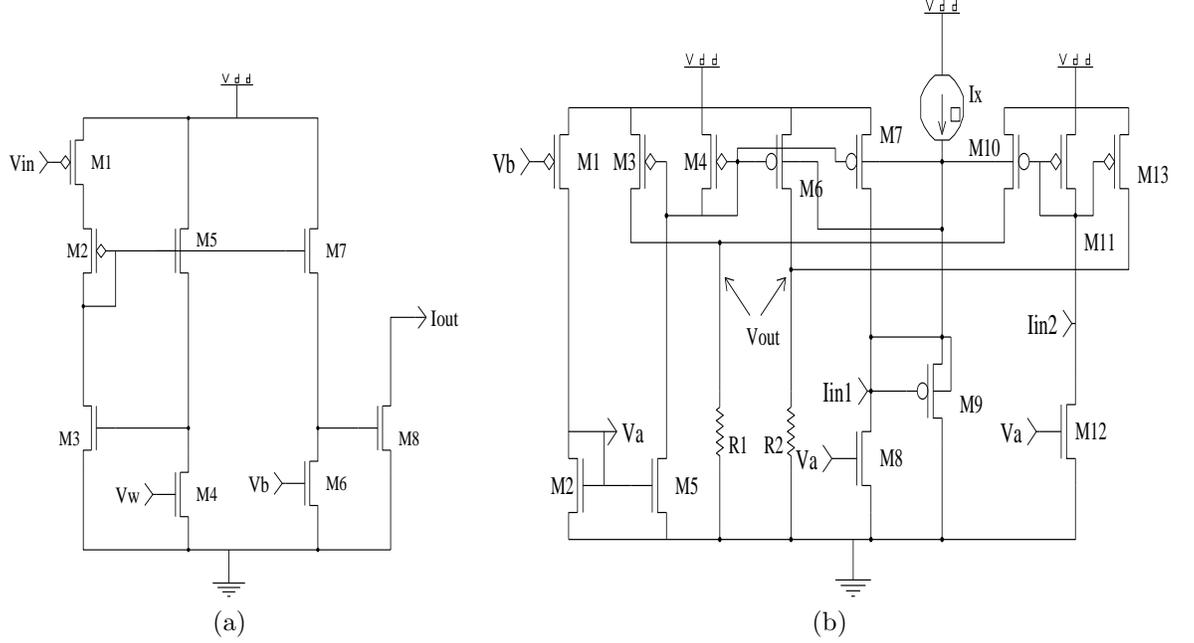


Figure 2.9: One and four-quadrant current mode multipliers. (a) One quadrant current mode weight multiplier. (b) Four quadrant current mode multiplier.

by the currents  $I_{M3}$  and  $I_{M10}$  as follows:

$$I_{R1} = I_{M3} + I_{M10} = I_b + (I_{in1} + I_b)(I_{in2} + I_b)/I_b \quad (2.73)$$

and the current flowing through the resistor  $R_2$  is  $I_{R2}$  and can be found by

$$I_{R2} = I_{M6} + I_{M13} = I_{M7} + I_{M11} = (I_{in1} + I_b) + (I_{in2} + I_b) \quad (2.74)$$

The difference between these currents is

$$I_{R1} - I_{R2} = \frac{I_{in1}I_{in2}}{I_b} \quad (2.75)$$

Assuming  $R_1 = R_2$ , then the output voltage  $V_{out}$  can be represented in terms of the input currents as follows:

$$V_{out} = R(I_{R1} - I_{R2}) = R \frac{I_{in1}I_{in2}}{I_b} \quad (2.76)$$

In the implementation of the four quadrant multiplier, we used diode connected transistors to realize  $R_1$  and  $R_2$ .

## Chapter 3

# Monolithic Implementations of Analog VLSI Motion Sensors

In this work, we present two different visual motion sensors that are implemented by making use of the monolithic design strategy. We describe their algorithms and circuit level implementations. These chips were fabricated in a standard  $1.5\mu\text{m}$  CMOS process and the die size used for all the chips is  $2.1 \times 2.1\text{mm}^2$ . The MOSFETs involved in the computational stages of the models operate in the subthreshold region to minimize the power consumption of the sensors. First, we present direction and velocity sensors realized by a hysteretic winner-take-all based algorithm. Second, we introduce a nonlinear differentiator based direction sensor.

### 3.1 Hysteretic winner-take-all based motion sensor

Motion computation methods are classified into feature tracking and intensity based algorithms. Feature tracking algorithms extract spatial or temporal features in an optical image and track these features to estimate their velocity. Intensity based algorithms utilize image irradiance directly to compute the optical flow field. In this study, we present a novel algorithm that incorporates the capabilities of feature tracking based methods in order to perform direction and interpixel transit time computations.

Barlow and Levick [5] explained a mechanism for the direction selectivity found in the rabbit retina. This retina requires inhibitory connections to achieve direction selectivity as shown in Figure 3.1. In this mechanism when a moving edge passes over two neighboring photoreceptors in the preferred direction from left to right, the left photoreceptor is excited first, causing its direction selective (DS) cell to fire and the right DS cell to be inhibited. After the edge reaches the right photoreceptor, an inhibitory connection between the right photoreceptor and left DS cell retards further output from the left DS cell. In the other case, when an edge is moving in the null direction (right to left), the activated right photoreceptor causes the right DS cell to fire and retards the left DS cell from firing. Each DS cell fires only when there is a motion in the direction that it is sensitive to. Delbruck and Benson [7] implemented a direction selective silicon retina by utilizing this mechanism. Basically, their implementation tracks the temporal features in the image and uses them to inhibit the neighboring cells to find the direction of stimulus.

In the present work, we show that the null inhibition mechanism that is inherent in the winner-take-all neural network can be used to realize the Barlow-Levick model and to compute an optical flow field without requiring any temporal differentiation. In this way we obtain a robust motion sensor that is independent of spatial and temporal frequency of the stimuli in the range of frequencies determined by the frequency response of the photoreceptor.

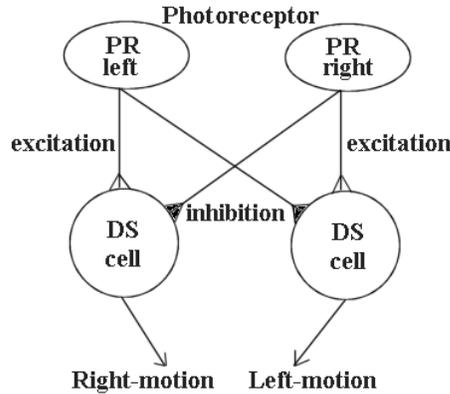


Figure 3.1: The Barlow and Levick model of direction selectivity.

### 3.1.1 Algorithm of the sensor

This motion computation algorithm is based on the idea that temporal changes in intensity levels sensed by photoreceptors can be tracked to compute optical flow created by image motion. In case of a positive or negative intensity change in two neighboring photoreceptors, direction information can be extracted from the optical image by determining the first activated receptor. Once it is determined, the challenge is to be able to inhibit the neighboring photoreceptor till the intensity changes in both receptors go below some predetermined value or change their signs.

To be able to realize such a scheme, a method that is independent of the spatiotemporal frequency of the stimulus should be developed. The algorithm illustrated in Figure 3.2 can potentially solve this problem by storing the information of the first excited photoreceptor and inhibiting the neighboring photoreceptor with the sum of the intensity changes of two photoreceptors.

Photoreceptors are arranged in such a way that there is always a  $\Delta x$  spatial distance between adjacent photoreceptors. In turn, this arrangement ensures that any image motion creates an excitation in a photoreceptor with a time difference compared to the excitation in its neighboring photoreceptor. The analog VLSI implementation of the motion algorithms discretize space into pixels and this helps to simplify the motion computation performed in each pixel. By using interpixel transit time information we can extract the motion information from the changes in illumination levels of the optical image.

The algorithm shown in Figure 3.2 extracts intensity changes by comparing the photoreceptor outputs to their long term running averages. These temporal changes in illumination levels are amplified and then half-wave-rectified, because we are only interested in one type of intensity change. After that we determine the first activated photoreceptor by detecting the first intensity change bigger than the predetermined threshold value. For this purpose we exploit the winner-take-all idea to find the bigger response and inhibit the smaller one (Section 2.7). Once we are able to find the first activated photoreceptor we use this information to set the winner state to this receptor and to inhibit the neighboring rectifier output. If there is motion information in this intensity change, then the neighboring receptor will respond in the same way with a reasonable phase lag. As a result, the loser will be the second receptor whose response exceeds the threshold after the first one does. In order to make sure that the direction computation output of this intensity change stays the same in the interval where at least one of two photoreceptor outputs is above threshold, we award the sum of the rectified signals to the winner. In this way, intensity changes in two neighboring photoreceptors will result in one direction output. In order to make sure that the winner and loser states are computed again after the photoreceptors adapt to intensity changes or the two rectified signals go below the predetermined threshold value, the algorithm sets both rectified signals the loser.

The functioning of the algorithm in signal level is illustrated in Figure 3.3. In this figure, two

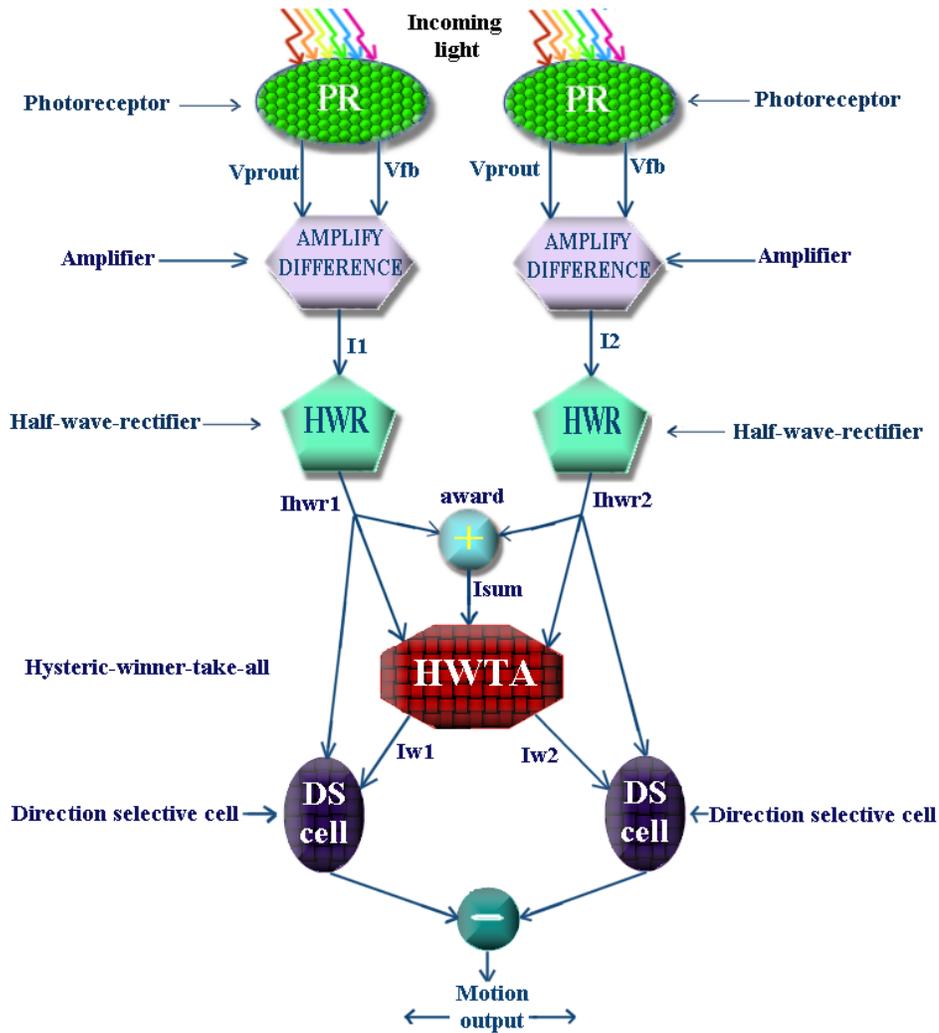


Figure 3.2: Hysteric winner-take-all based elementary motion detector.  $V_{prout}$  is the output of the photoreceptor and  $V_{fb}$  is its feedback voltage, which represents the adapted background light level of the environment observed by the photoreceptor.

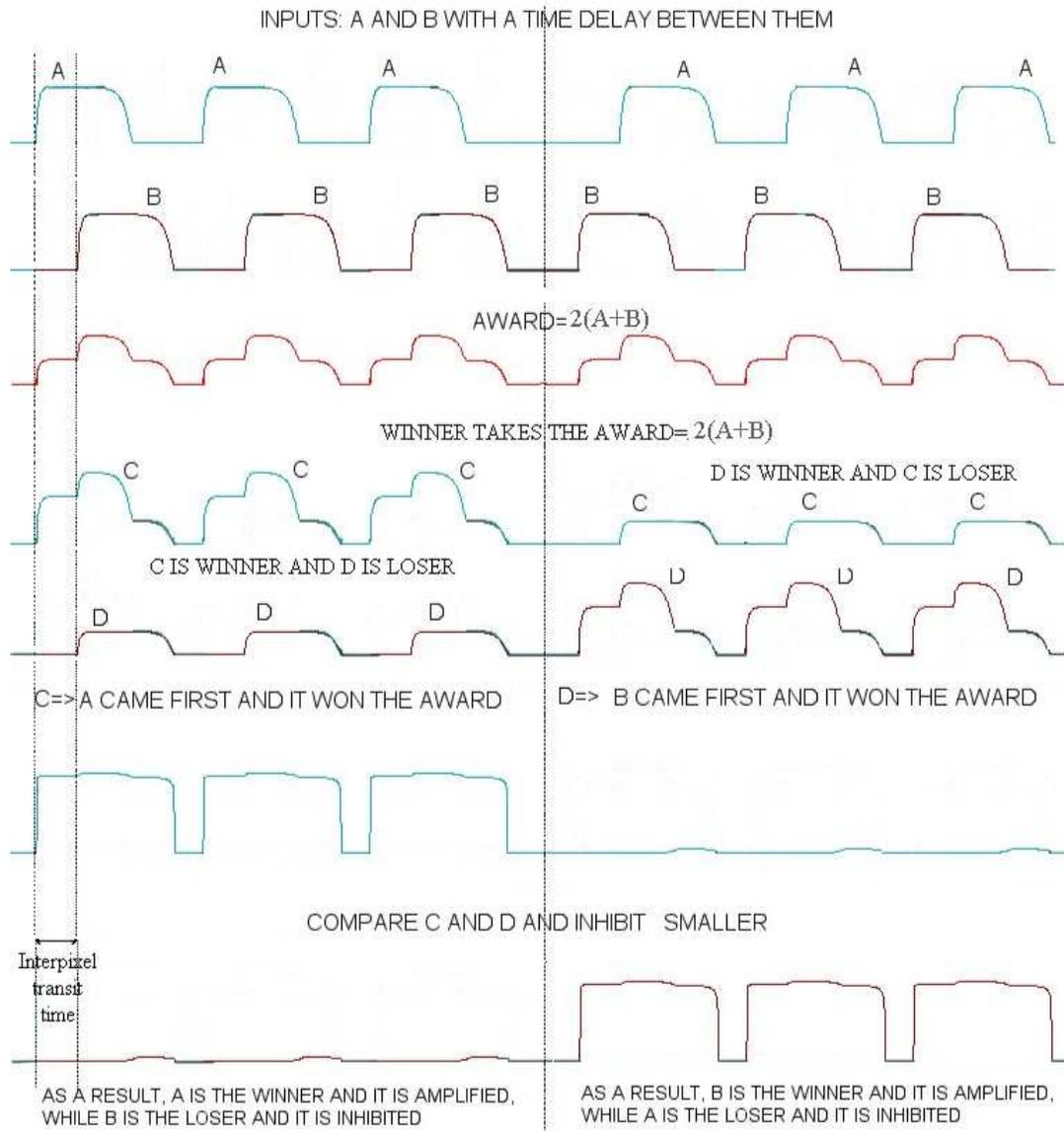


Figure 3.3: Functioning of the hysteretic winner-take-all based motion detection algorithm in signal level.

cases in which either photoreceptor A or B is excited before the other one are illustrated. In the case when A comes first, the sum of A and B is acquired by A and the total result becomes the signal C. The difference between signals C and D is used to inhibit the loser and excite the winner. Thus, one of the signals is always suppressed by the system and this helps us obtain a direction selectivity to image motion.

### 3.1.2 Transistor-level implementation of the algorithm

The algorithm is implemented in two different versions to show the diversity of implementations that can be obtained by employing the hysteretic winner idea. All these implementations contain four different circuit blocks, namely, a photoreceptor, a transconductance amplifier, a half wave rectifier and a hysteretic winner-take-all circuit.

The transistor-level implementation of the first version of the algorithm is shown in Figure 3.4. This circuit computes the direction of motion and can be used to find direction vectors in the optical flow field. The amplitude of the final output of the circuit depends on the contrast level of stimuli. In circuit simulations, the adaptive photoreceptor by Delbruck [14] is used, whereas, in the final implementation of the algorithm (Figure A.3), the photoreceptor by Liu [68] is employed. The adaptation time constant of this photoreceptor circuit can be controlled by an external bias (see Section 2.2). Such control is needed since the adaptation rate determines the amplitude and time response of the circuit. For short timescales, if we ignore the DC offsets of the adaptive photoreceptor outputs then it can be written that  $V_{prouit} = \frac{C_1+C_2}{C_1} V_{fb}$  (Equation 2.9) and  $V'_{fb} = \frac{V_T}{\kappa} \ln(\frac{I_{in}}{I_0})$  (Equation 2.12).

In addition to the adaptive photoreceptors, transconductance amplifiers are used in the implementation to remove the offset in photoreceptor voltage output and amplify the difference between the adaptive photoreceptor output and its feedback voltage. The feedback voltage adapts to the background illumination level and therefore the difference between these voltage levels indicates an intensity change in the relative contrast level of the input signal. The transconductance amplifier has a linear region to represent the current levels in case of small voltage differences and for reasonably large differences its response saturates. Accordingly, the current output of the transconductance amplifier to changes in a photoreceptor can be shown to be as follows:

$$I_{trout} = I_{bias} \tanh\left(\frac{\kappa}{2V_T}(V_{fb} - V_{prouit})\right) \quad (3.1)$$

For short timescales,

$$I_{trout} = I_{bias} \tanh\left(\frac{C_2}{2C_1} \ln\left(\frac{I_{in}}{I_0}\right)\right) \quad (3.2)$$

where  $I_{bias}$  is the drain-to-source current that is set by  $V_{diffbias}$  and  $I_{in}$  is the current driven by the photodiode. For long timescales since  $V_{prouit} = V_{fb}$ ,

$$I_{trout} = 0 \quad (3.3)$$

After this stage, a half wave rectifier is employed to extract only one type of change in intensity level relative to the background light intensity level. This situation is depicted for the rectified signals in Figure 3.5. Although half-wave-rectification seems like wasting half of the input signal information, in normal conditions if the adaptation time of the photoreceptors is adjusted properly one can always obtain a photoreceptor response that varies around the background intensity level for all kinds of stimuli. Assuming  $I_{out}$  is the current output of the transconductance amplifier, the response of the half wave rectifier in terms of photoreceptor voltages can be formulated as follows:

$$I_{hwr} = \begin{cases} |I_{bias} \tanh(\frac{C_2}{2C_1} \ln(\frac{I_{in}}{I_0}))| & \text{if } I_{trout} < 0 \\ 0 & \text{if } I_{trout} > 0 \end{cases}$$

The next stage of the circuitry is the implementation of the hysteretic winner-take-all circuit.

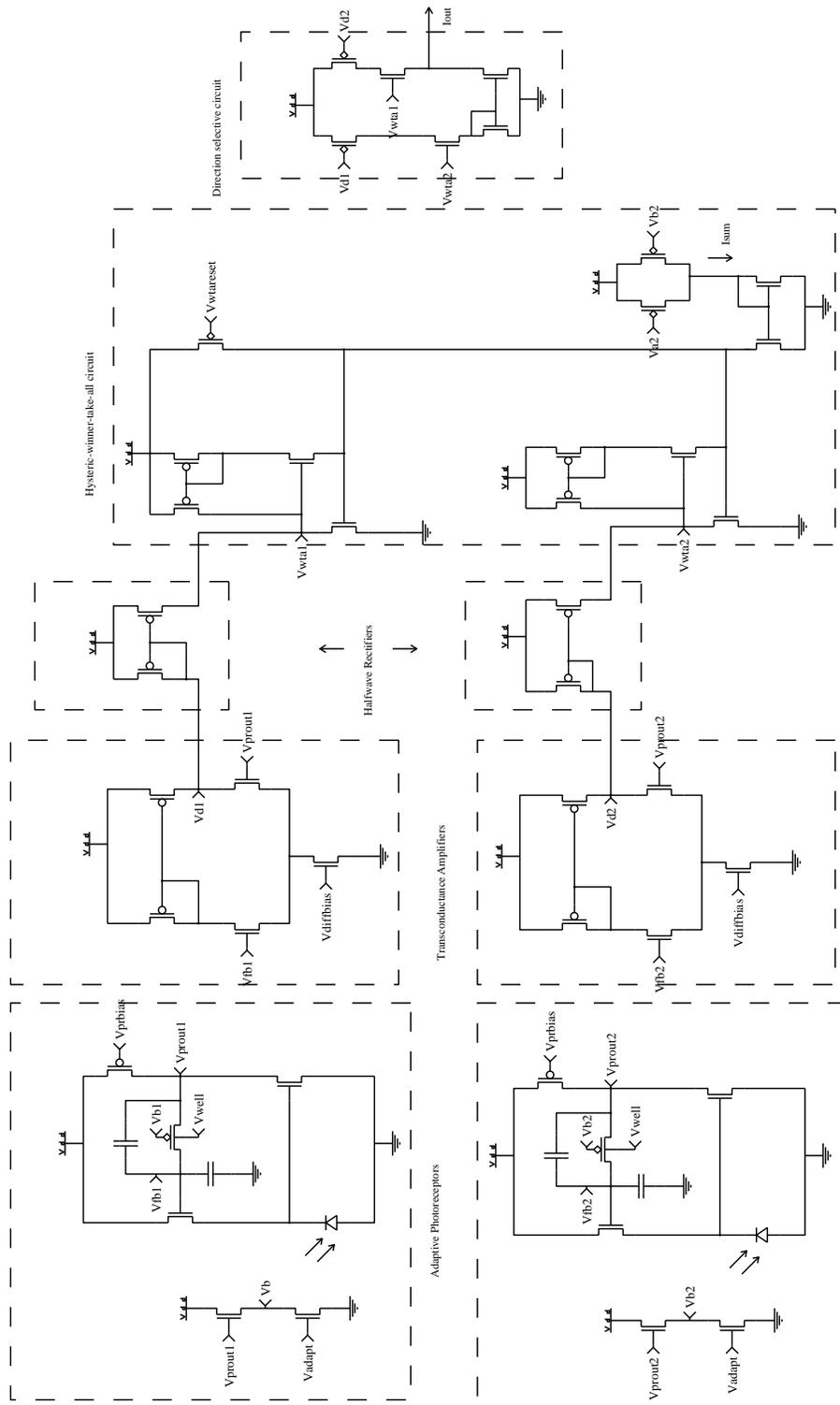


Figure 3.4: The first version of the transistor-level implementation of the hysteresis-based motion algorithm.

In its operation, a two-input hysteretic winner-take-all circuit compares two inputs, sets the winner state to the bigger input and then awards a predetermined bias to the winner [18] (see Section 2.7 for details). This circuit is changed (Figure 3.4) to meet the expectations specified in the algorithm. Firstly, a fixed third input,  $V_{wtareset}$  is added to the circuit as a reset and threshold bias in order to compare it with two other input signals and determine which of two transduced signals  $V_{d1}$  and  $V_{d2}$  is the first coming signal. Secondly, the fixed bias is altered to two times of the summation of the half-wave rectified signals, and in this way the status of the winner will be reserved till both inputs go below the threshold level. As a result of this operation, we will obtain a loser and a winner, and the information of winner and loser states as is depicted in Figure 3.6. By using Equation 2.67 and 2.68, the output of this circuit and its conditions can be found. For instance, when  $(V_{dd} - V_{wtareset}) > (V_{dd} - V_{d1}), (V_{dd} - V_{d2})$ , then

$$V_{wta1} = V_{wta2} = 0 \quad (3.4)$$

When  $(V_{dd} - V_{d1}) > (V_{dd} - V_{wtareset})$  before  $(V_{dd} - V_{d2}) > (V_{dd} - V_{wtareset})$ ,

$$V_{wta1} = \frac{V_T}{\kappa} \ln\left(\frac{I_{hwr1}}{I_0}\right) + \frac{V_T}{\kappa} \ln\left(\frac{2I_{hwr1}}{I_0}\right) = 2V_T \ln\left(\frac{\sqrt{2}I_{hwr1}}{I_0}\right), \quad V_{wta2} = 0 \quad (3.5)$$

After sometime  $(V_{dd} - V_{d1}) > (V_{dd} - V_{wtareset})$ , then

$$V_{wta1} = V_T \ln\left(\frac{I_{hwr1}}{I_0}\right) + V_T \ln\left(\frac{2(I_{hwr1} + I_{hwr2})}{I_0}\right) = V_T \ln\left(\frac{2I_{hwr1}(I_{hwr1} + I_{hwr2})}{I_0^2}\right), \quad V_{wta2} \approx 0 \quad (3.6)$$

Later when  $(V_{dd} - V_{d1}) < (V_{dd} - V_{wtareset})$  and  $(V_{dd} - V_{d2}) > (V_{dd} - V_{wtareset})$ ,

$$V_{wta1} = V_T \ln\left(\frac{I_{hwr2}}{I_0}\right), \quad V_{wta2} \approx 0 \quad (3.7)$$

Lastly, when  $(V_{dd} - V_{wtareset}) > (V_{dd} - V_{d1})$  and  $(V_{dd} - V_{d2})$ , the circuit resets itself and goes back to its initial condition where

$$V_{wta1} = V_{wta2} \approx 0 \quad (3.8)$$

In the other case when the first signal that goes below  $V_{wtareset}$  is  $V_{d2}$ , then the result can be found by interchanging  $V_{wta1}$  and  $V_{wta2}$ .

In the final stage, the winner and loser voltages are used to sample the rectified currents of the intensity changes in the both photoreceptor outputs. As seen in Figure 3.4, the output voltage of the first candidate of the winner state  $V_{wta1}$  is used to sample the second rectified signal  $V_{d2}$  and the second candidate,  $V_{wta2}$ , is used to sample the first rectified signal  $V_{d1}$ . In other words, the early excitation in one of two neighboring photoreceptors causes an inhibition to its rectified signal and excitation to the rectified signal of the neighboring photoreceptor. The reason for this kind of sampling is that we are trying to ensure that the sampled signal is the output of a moving stimulus in the visual field of the photoreceptors with a certain time delay of transition between each receptor. In this way, it is guaranteed that a sudden change in a photoreceptor output does not produce a motion output in its interaction with a neighboring photoreceptor if it does not have a motion component in that direction. In addition, if the competition between two neighboring photoreceptors for the winning state cannot be concluded, which means that the velocity of the stimulus is too fast or the orientation of the stimulus does not have any motion component in that arrangement of photoreceptors (but excites both photoreceptors), then both inputs are set to the winning state and the output of the circuit becomes zero. As a result, we can conclude that the magnitude of the current output of the overall circuit illustrated in Figure 3.7 can be formulated as follows:

$$|I_{out}| = |I_{bias} \tanh\left(\frac{C_2}{2C_1} \ln\left(\frac{I_{in}}{I_0}\right)\right)| \quad (3.9)$$

Different circuit implementations of this algorithm can be used to exploit the hysteretic winner idea. For example, in the second version of circuit implementation (Figure 3.8), two more transconductance amplifiers are used for each pixel in order to sample the winning output at the right time. If this circuit is analyzed it can be seen that the duration of current output, which is the same as the interpixel transit time, is proportional to the inverse of the velocity. For instance, when the first signal is the winner and the second one is the loser (Figure 3.9), one can observe that  $V_{out1} \gg V_{out2}$  in the specified time interval, because the first input is amplified and the second one is inhibited. The transconductance amplifiers will provide (for reasonable contrast levels) a time window by which we can compute the interpixel transit time of an edge passing over the photoreceptors. This information can be utilized to obtain the velocity of the stimulus if one computes the reciprocal of the duration of the output. Considering the functioning of the circuitry, the current output can be formulated.

Firstly, since  $I_{hwr1} = |I_{bias} \tanh(\frac{C_2}{2C_1} \ln(\frac{I_{in1}}{I_0}))|$  and is supplied to the second transconductance amplifier as a bias, the result of this operation will be a multiplication that can be shown as follows:

$$I_{mulout} = |I_{bias} \tanh(\frac{C_2}{2C_1} \ln(\frac{I_{in1}}{I_0}))| \times \tanh(\frac{\kappa}{2V_T} (V_{fb2} - V_{prou2})) \quad (3.10)$$

$$I_{mulout} = |I_{bias} \tanh(\frac{C_2}{2C_1} \ln(\frac{I_{in1}}{I_0}))| \times \tanh(\frac{C_2}{2C_1} \ln(\frac{I_{in2}}{I_0})) \quad (3.11)$$

Since the half-wave rectified output is zero when  $V_{fb1} < V_{prou1}$  or  $V_{fb2} < V_{prou2}$ ,  $I_{mulout}$  also becomes zero. In other cases  $I_{mulout}$  can be shown to be as follows:

$$I_{mulout} = |I_{bias} \tanh(\frac{C_2}{2C_1} \ln(\frac{I_{in}}{I_0}))| \times |\tanh(\frac{C_2}{2C_1} \ln(\frac{I_{in2}}{I_0}))| \quad (3.12)$$

Secondly, the current response of the overall circuit can be found by integrating the effect of the last transconductance amplifier with the above equation. Hence, the final output is

$$I_{out} = |I_{bias} \tanh(\frac{C_2}{2C_1} \ln(\frac{I_{in}}{I_0}))| \times |\tanh(\frac{C_2}{2C_1} \ln(\frac{I_{in2}}{I_0}))| \times \tanh(\frac{\kappa}{2V_T} (V_{wta1} - V_{wta2})) \quad (3.13)$$

where the last expression that includes hysteretic winner-take-all outputs can be written in terms of input currents for the conditions indicated in Equations 3.4, 3.5, 3.6, and 3.7.

In this sense, the hysteretic winner-take-all based algorithm can be utilized not only to obtain a direction sensor but also a velocity sensor. Besides, the circuit shown in Figure 3.8 is useful to decrease any error that might be caused by sudden intensity fluctuations in the focal plane of the sensor. The rectified output of the first transconductance amplifier is used as an input to the second transconductance amplifier, which detects the changes in the signal level of the second photoreceptor. This operation corresponds to a multiplication of the two half-wave rectified signals as shown in Equation 3.12. Hence, this will make sure that we will obtain a motion output when there are activities in both photoreceptors and also when there is enough time difference in the transition of the stimulus so that the winning signal can be determined. As is the nature of the winner-take-all circuit, if the ratio of the input signals cannot exceed the predetermined ratio of inputs ( $I_1/I_2$ ) in the race for the winner state, then both inputs will be assigned as winner, so zero motion output will be extracted in this case.

In conclusion, this algorithm can be realized as a direction sensor by the first version of the circuit implementation (Figure 3.4) with 1 photodiode, 31 transistors and 2 capacitors to detect the direction of motion. The simulation result of this implementation (Figure 3.7) shows clearly the direction selectivity of the sensor. The second version of the circuit implementation of the algorithm (Figure 3.8) includes 1 photodiode, 36 transistors and 2 capacitors to compute the interpixel transit time, which is in turn the indication of velocity. Further simulation results of this implementation (Figure 3.10) demonstrate that when the frequency of the input signal is decreased from 50Hz to 10Hz, the interpixel transit time increases 5 times. The layouts of the final chip and two different

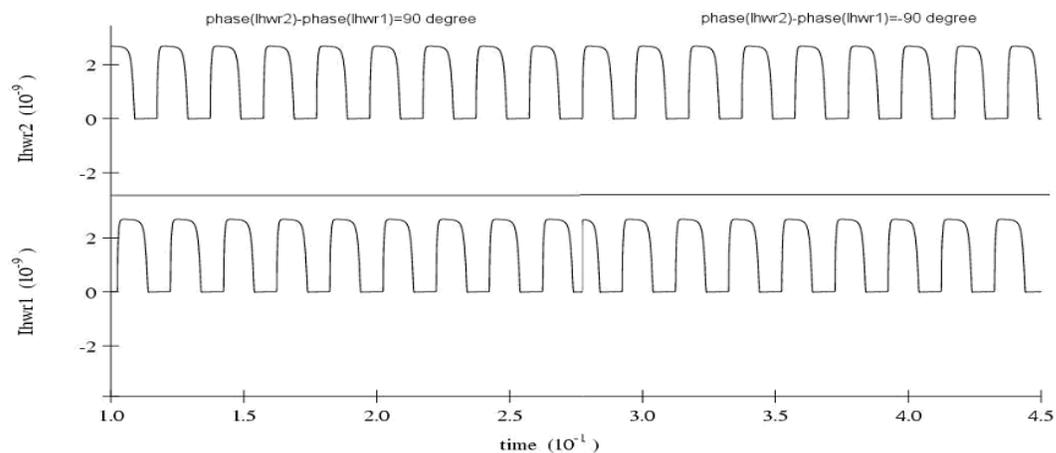


Figure 3.5: Response of two half wave rectifiers, which are used to rectify the responses of transconductance amplifiers to the sinusoidal 50Hz gratings with 90-degree phase difference.

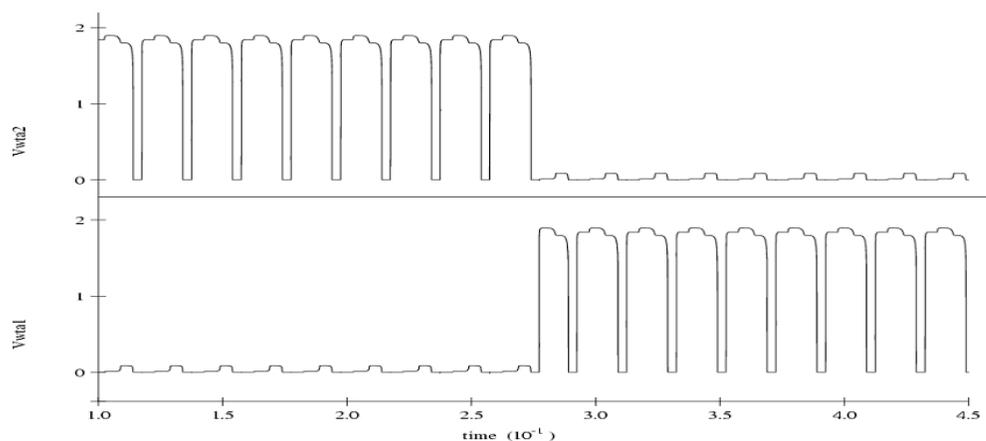


Figure 3.6: Responses of the hysteretic winner-take-all circuit. Once the first incoming signal is determined, the winner is amplified and loser is inhibited.

versions of the implementations of the pixels are shown in Figures A.1, A.2, and A.3. Unfortunately, the characterization results of this chip are not yet available.

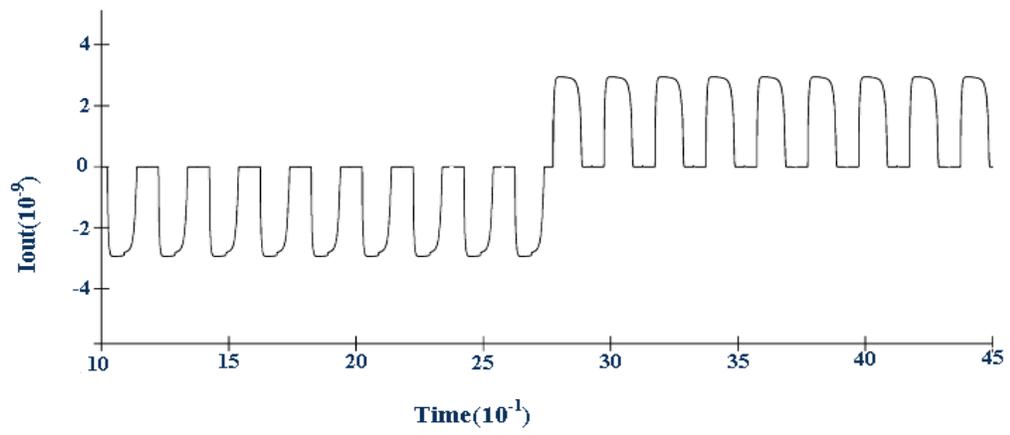


Figure 3.7: The output of the first version of the circuit is obtained by sampling the loser of the competition by the winner output.

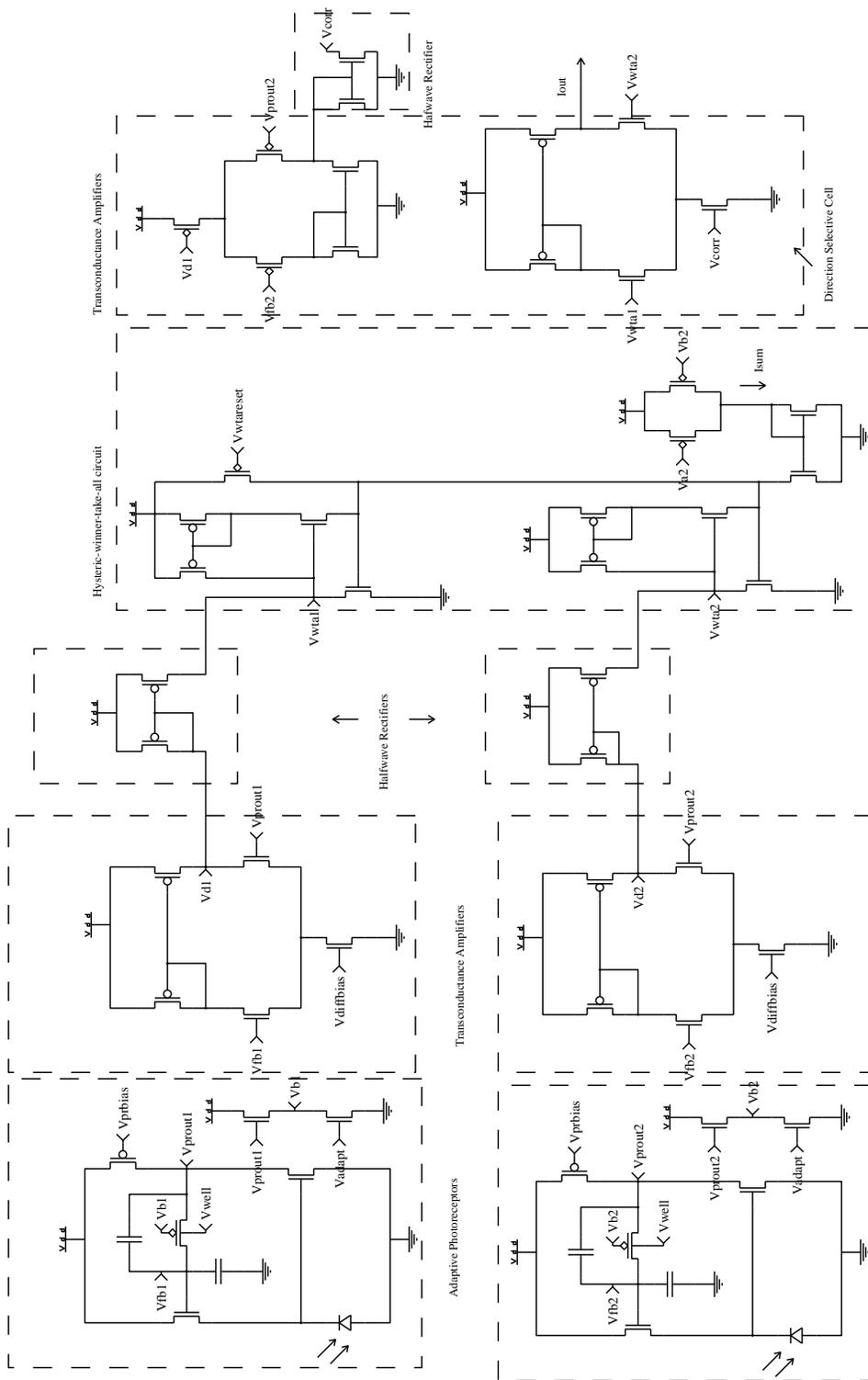


Figure 3.8: The second version of the circuit implementation of the hysteric winner-take-all based motion algorithm.

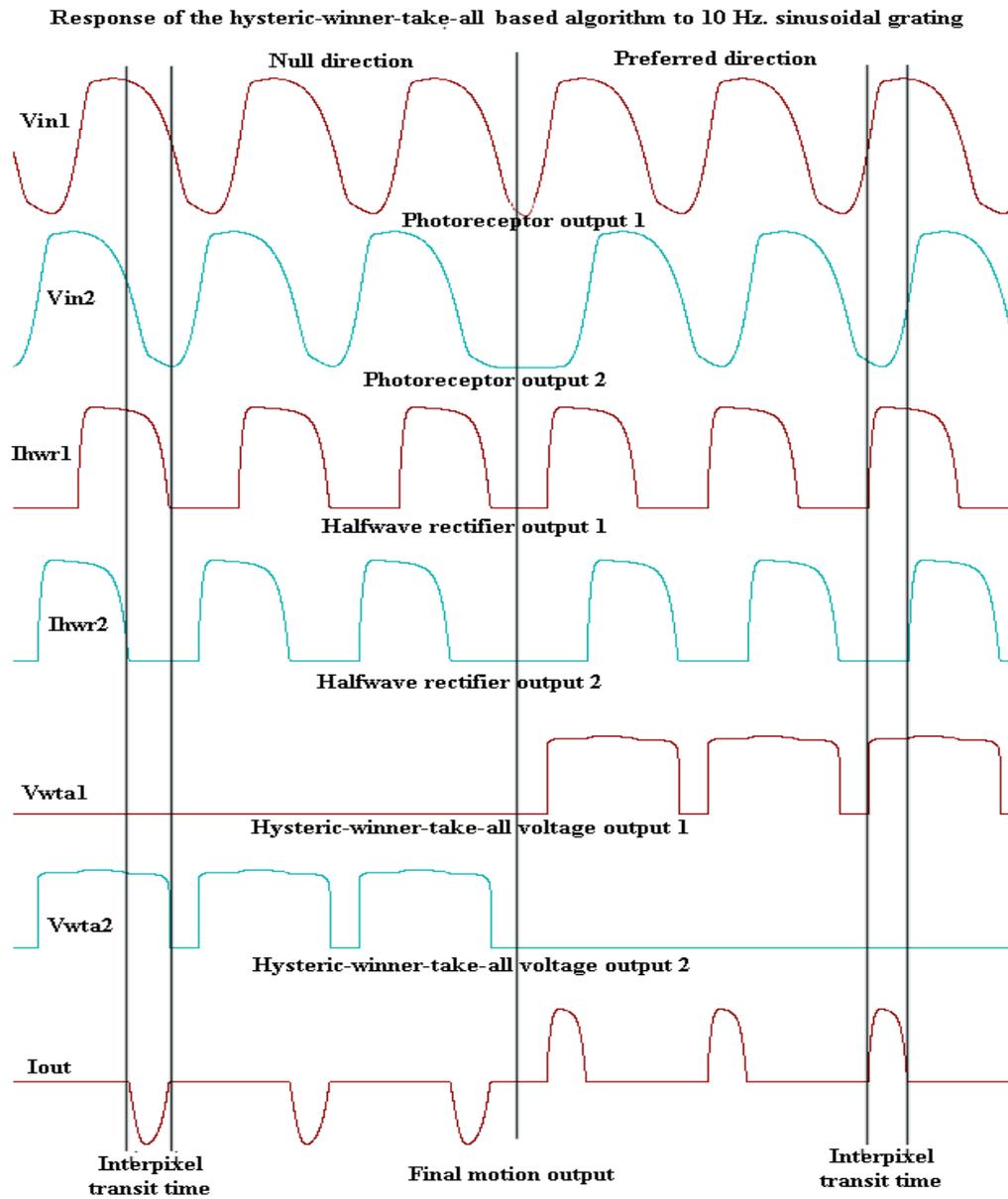


Figure 3.9: The final output and the computation performed by the second version of the circuit for 10Hz sinusoidal grating input.

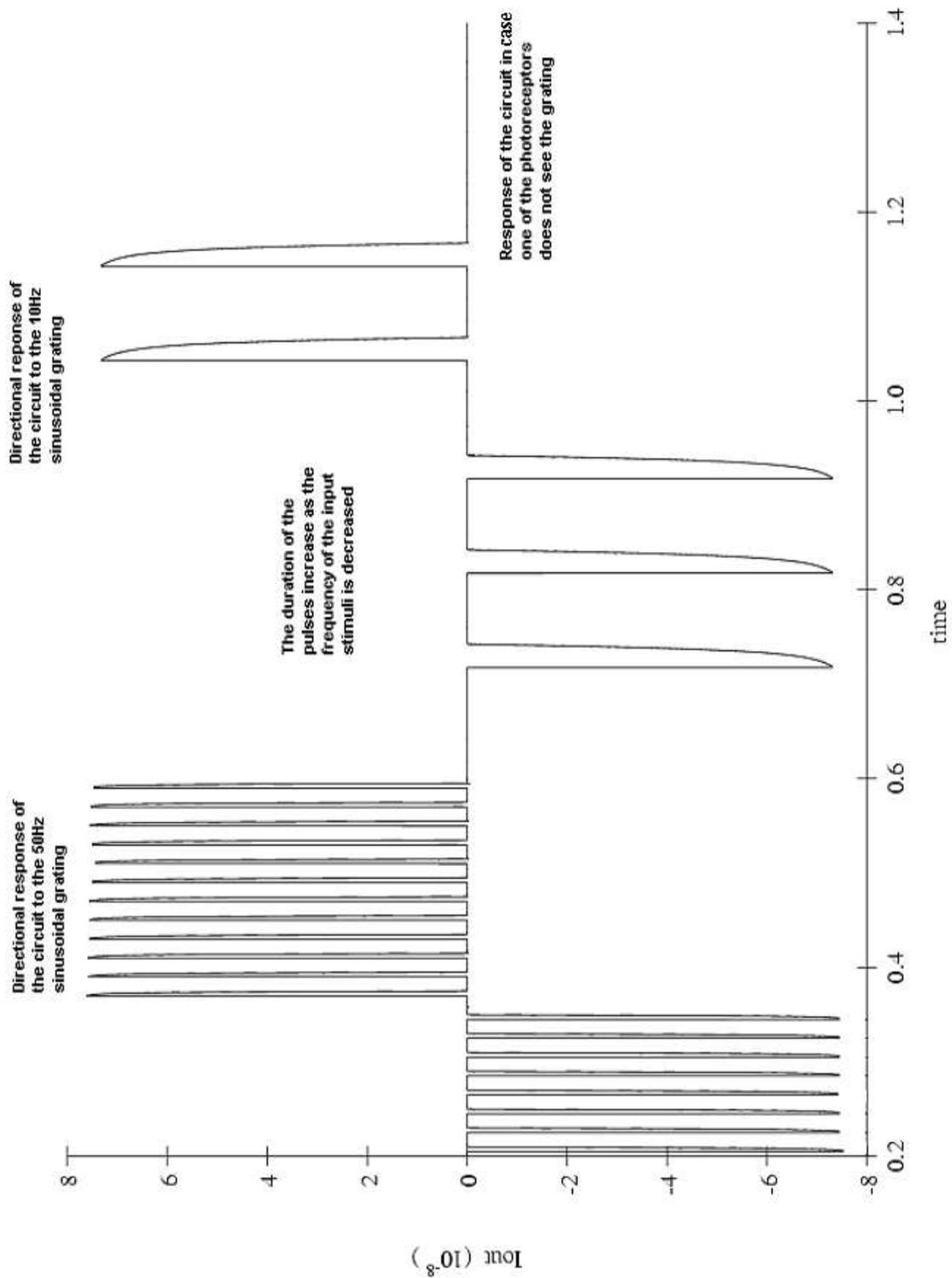


Figure 3.10: The final output of the second version of the circuit. It is simulated with 50Hz and then 10Hz sinusoidal grating input. The interpixel transit time increases 5 times as the temporal frequency goes from 50Hz to 10Hz.

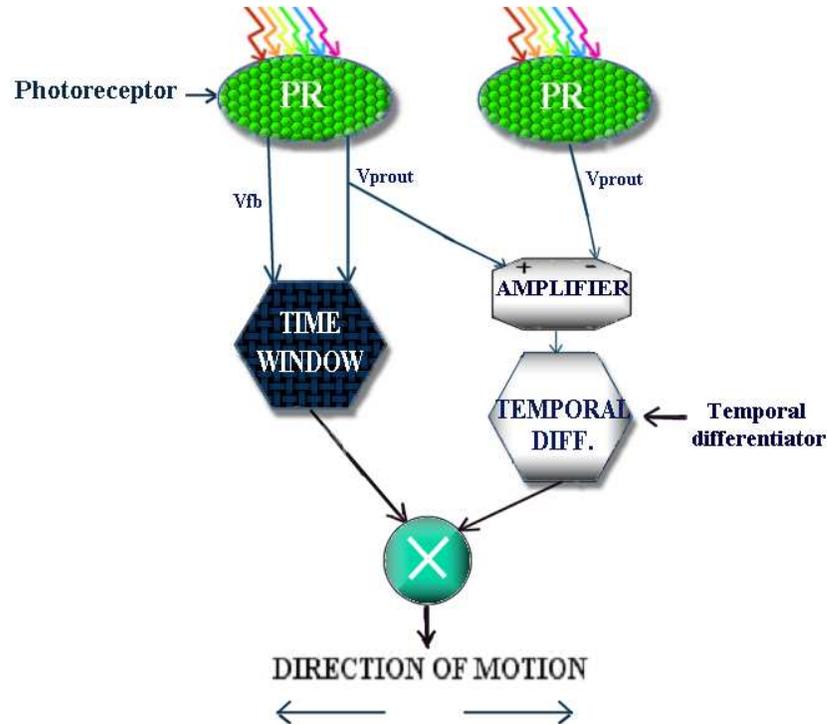


Figure 3.11: The nonlinear differentiator based direction sensor.  $V_{prout}$  is the output of the photoreceptor and  $V_{fb}$  represents the adaptation state of the photoreceptor.

## 3.2 Nonlinear differentiator based direction sensor

In this study, we use temporal changes in the amplified difference of two neighboring photoreceptor outputs to find their crossing point instead of their edges to determine the direction of image motion. Intensity changes in two neighboring photoreceptors can be different from each other, but we assume that if the illumination changes in these receptors contain motion information then these receptor outputs must cross or have the same value at some point.

Temporal differentiation of the amplified difference of input signals gives its strongest response when these signals cross. In other words, we focus on the changes in the states of neighboring photoreceptor outputs. In this study we use the term ‘state’ to describe the level of a photoreceptor output relative to its neighboring photoreceptor output. Hence, if the photoreceptor output is bigger than its neighboring photoreceptor output, then the state is defined as ‘ON’ and if it is less than the neighboring photoreceptor output then the state is defined as ‘OFF’.

By using the transitions from ON-to-OFF and OFF-to-ON we can obtain spike/pulse like responses from temporal differentiation. These responses can be utilized in motion computation by multiplying them with a time-window generated by the difference between a photoreceptor output and its adaptation state.

### 3.2.1 Algorithm of the sensor

In this section we explain the algorithm of the direction sensor shown in Figure 3.11. As is illustrated in the algorithm, the temporal differentiation of the amplified spatial difference of two neighboring photoreceptors can be utilized to detect the direction of the motion. The idea of this algorithm is developed by making use of the computational properties of visual motion perception.

Within a specified time window, the difference between neighboring photoreceptor outputs can

be differentiated to acquire information on their relative changes. In the existence of motion, such as with a sinusoidal grating input, these receptors yield outputs with a phase difference relative to each other and this phase difference can be used to determine the leading and lagging signals within the time interval of positive or negative illumination change in one of the neighboring photoreceptors. In other words, the slope of the difference of these signals will change its sign once in the interval of one cycle of a sinusoidal grating. Therefore, in the mean, the multiplication of the slope and intensity change gives the direction of motion. In this section we prove this for a sinusoidal grating input.

Assume that two neighboring photoreceptors are excited with one-dimensional sinusoidal grating inputs and  $w_t$  is the temporal frequency and  $w_x$  is the spatial frequency of this grating. If the mean luminance of the signal is  $I$ , then we can write the output of these photoreceptors as follows:

$$I_1 = I + \Delta I \sin(w_t \cdot t + w_x \cdot x) \quad (3.14)$$

$$I_2 = I + \Delta I \sin(w_t \cdot t + w_x \cdot x \pm w_x \cdot \Delta x) \quad (3.15)$$

where  $\Delta x$  corresponds to the distance between neighboring photoreceptors, and plus and minus indicate the direction of motion. The aim is to compute the difference of these two signals and differentiate the difference of these signals.

$$I_1 - I_2 = \Delta I \cdot [\sin(w_t \cdot t + w_x \cdot x) - \sin(w_t \cdot t + w_x \cdot x \pm w_x \cdot \Delta x)] \quad (3.16)$$

$$\frac{\delta(I_1 - I_2)}{\delta t} = \Delta I \cdot w_t [\cos(w_t \cdot t + w_x \cdot x) - \cos(w_t \cdot t + w_x \cdot x \pm w_x \cdot \Delta x)] \quad (3.17)$$

By using the trigonometric identity  $\sin(a) - \sin(b) = 2 \cos(\frac{a+b}{2}) \sin(\frac{a-b}{2})$ ,

$$\frac{\delta(I_1 - I_2)}{\delta t} = -2\Delta I \cdot w_t \cdot (\sin(w_t \cdot t + w_x \cdot x \pm \frac{w_x \cdot \Delta x}{2}) \sin(\mp \frac{w_x \cdot \Delta x}{2})) \quad (3.18)$$

In the time domain, the first sine term is the only effective term that determines the change of the sign. Hence, we can have only two formulas for the derivative of difference.

For the preferred direction:

$$\frac{\delta(I_1 - I_2)}{\delta t} = 2\Delta I \cdot w_t \cdot [\sin(w_t \cdot t + w_x \cdot x - \frac{w_x \cdot \Delta x}{2}) \sin(\frac{w_x \cdot \Delta x}{2})] \quad (3.19)$$

For the opposite direction:

$$\frac{\delta(I_1 - I_2)}{\delta t} = -2\Delta I \cdot w_t \cdot [\sin(w_t \cdot t + w_x \cdot x + \frac{w_x \cdot \Delta x}{2}) \sin(\frac{w_x \cdot \Delta x}{2})] \quad (3.20)$$

As seen from the formulas, the direction computation can be reduced to composing the time window by which the sign of the first sine term can be changed in the interval  $0^\circ < \Theta < 180^\circ$  or  $180^\circ < \Theta < 360^\circ$  (where  $\Theta$  represents the angle of the sine term) according to the direction of motion. After removing its offset the first input signal can be used as a time window. Then the expression shown below can be obtained by multiplying the time window by the derivative. The output of this computation is depicted in Figure 3.12 for preferred direction, no motion and null direction cases.

$$\Delta I \cdot \sin(w_t \cdot t + w_x \cdot x) \cdot \frac{\delta(I_1 - I_2)}{\delta t} =$$

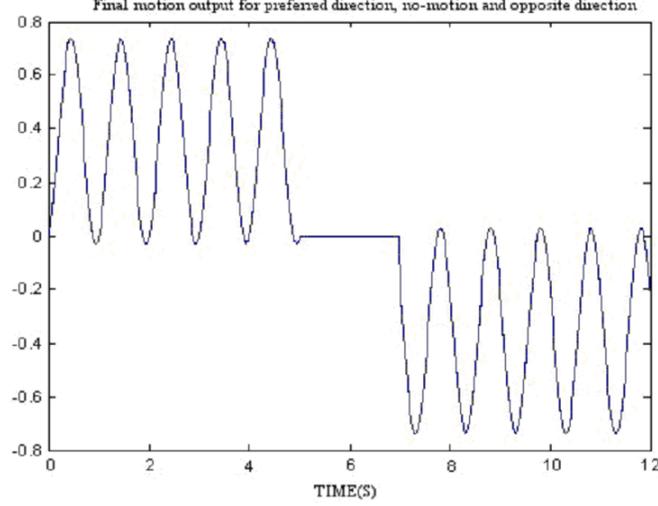


Figure 3.12: The result of Equation 3.24 when a normalized sinusoidal grating with 45-degree phase difference is used to simulate preferred direction, no-motion and null direction cases respectively.

$$= -2\Delta I^2 \cdot w_t \cdot [\sin(w_t \cdot t + w_x \cdot x) \cdot \sin(w_t \cdot t + w_x \cdot x \pm \frac{w_x \cdot \Delta x}{2}) \cdot \sin(\mp \frac{w_x \cdot \Delta x}{2})] \quad (3.21)$$

$$= \Delta I^2 \cdot w_t \cdot \sin(\mp \frac{w_x \cdot \Delta x}{2}) \cdot [\cos(2 \cdot (w_t \cdot t + w_x \cdot x) \pm \frac{w_x \cdot \Delta x}{2}) - \cos(\mp \frac{w_x \cdot \Delta x}{2})] \quad (3.22)$$

By using the trigonometric identity  $\cos(2\theta) = 1 - 2\sin^2(\theta)$ ,

$$I_{out} = \Delta I^2 \cdot w_t \cdot \sin(\mp \frac{w_x \cdot \Delta x}{2}) \cdot (1 - \cos(\frac{w_x \cdot \Delta x}{2})) \quad (3.23)$$

$$= -2\Delta I^2 \cdot w_t \cdot \sin(\mp \frac{w_x \cdot \Delta x}{2}) \sin^2(w_t \cdot t + w_x \cdot x \pm \frac{w_x \cdot \Delta x}{4})$$

$$I_{out} = 2\Delta I^2 \cdot w_t \cdot \sin(\mp \frac{w_x \cdot \Delta x}{2}) \cdot \sin^2(\frac{w_x \cdot \Delta x}{4})$$

$$= -2\Delta I^2 \cdot w_t \cdot \sin(\mp \frac{w_x \cdot \Delta x}{2}) \sin^2(w_t \cdot t + w_x \cdot x \pm \frac{w_x \cdot \Delta x}{4})$$

For  $\frac{w_x \cdot \Delta x}{4} \ll (w_t \cdot t + w_x \cdot x)$ , an approximation can be made to simplify the above equation. In this case,  $\sin^2(w_t \cdot t + w_x \cdot x \pm \frac{w_x \cdot \Delta x}{4})$  compared to  $\sin^2(\frac{w_x \cdot \Delta x}{4})$  plays the main role in determining the amplitude of the output. Therefore, the final expression can be approximated as follows:

$$I_{out} \approx 2\Delta I^2 \cdot w_t \cdot \sin(\pm \frac{w_x \cdot \Delta x}{2}) \sin^2(w_t \cdot t + w_x \cdot x \pm \frac{w_x \cdot \Delta x}{4}) \quad (3.24)$$

In this equation, the first sine term determines the direction of motion and its sign depends on the direction of the first activated signal. As a result, it can be concluded that one can easily obtain the direction of motion by just observing the sign of the differentiation in the time interval when the signal cycles between  $0^\circ < \Theta < 180^\circ$  and/or  $180^\circ < \Theta < 360^\circ$ . The motion output shown in Figure 3.12 confirms the obtained results.

In contrast to the gradient-based analog VLSI motion sensor developed by Deutschmann [16], this sensor uses the interaction of two pixels only. Besides, this sensor does not use a spatial derivative to calculate the direction of motion. Furthermore, the direction selectivity of the sensor can be improved more if at the last stage, a low-pass filter is employed. The low-pass filter not only decreases the level of noise, but also provides integration that is needed to compensate for the small phase difference between the time window and the differentiated signal.

### 3.2.2 Transistor-level implementation of the algorithm

This algorithm can be implemented in different ways. Depending on how the time window is obtained and how multiplication or sampling is achieved, different circuit blocks can be employed. However, in all implementations, photoreceptors (by Delbruck [14] for circuit simulations and by Liu [68] for the final implementation) and nonlinear differentiators are employed.

The first version of the circuit implementation (Figure 3.13) incorporates the adaptive characteristics of the photoreceptor to obtain a temporal window. The difference of neighboring photoreceptor outputs are amplified by a differential amplifier to be differentiated by a nonlinear differentiator. Using formulas 2.25 and 2.26, the output of the differential amplifier can be found.

When  $V_{prou1}$  is bigger than  $V_{prou2}$ , then the ON state is obtained and if it is less than  $V_{prou2}$  then the state becomes OFF. Therefore, the differential amplifier determines the timing of the nonlinear differentiator output when transitions from ON-to-OFF or OFF-to-ON occur. The nonlinear differentiator responds to these transitions with decaying pulses.

As explained in Section 2.6, for  $t \gg \frac{CV_T}{\kappa I_0} e^{-\frac{\kappa V_C(t=0)}{V_T}}$  (where C is the capacitance in the circuit, and  $V_C(t=0)$  is the initial voltage of the capacitor at  $t=0$ ), this circuit becomes independent of the initial capacitor voltage  $V_C(t=0)$ , and the output current becomes:

$$I_{nldout}(t) = \frac{CV_T}{\kappa t} e^{\frac{V_{gain}}{V_T}} \quad (3.25)$$

The amplitude of the currents obtained from  $V_{diff1}$  and  $V_{diff2}$  are amplified by  $V_{gain1}$  and  $V_{gain2}$ , respectively. After that a four quadrant multiplier (realized by two transconductance amplifiers) is employed to multiply the differentiation of the amplified difference of two neighboring photoreceptor outputs with the time window obtained from the first photoreceptor. This multiplier is different from the one described in Section 2.8 and uses two transconductance amplifiers to multiply the difference between voltage inputs  $V_{prou1}$  and  $V_{prou2}$  with the bias currents  $I_{diff1}$  and  $I_{diff2}$ . In this case the overall current output will be as follows:

$$I_{out}(t) = (I_{diff1}(t) - I_{diff2}(t)) \tanh\left(\frac{\kappa}{2V_T}(V_{prou}(t) - V_{fb}(t))\right) \quad (3.26)$$

This implementation uses 23 transistors and 3 capacitors per pixel. Deutschmann [16] implemented a similar circuit by utilizing three-pixel interaction, and in this study it is proved that direction information can be extracted by using only two photoreceptor interactions and without using a spatial differentiation. The simulation results prove that the circuit implementation of the algorithm computes the direction of motion unambiguously (Figure 3.14).

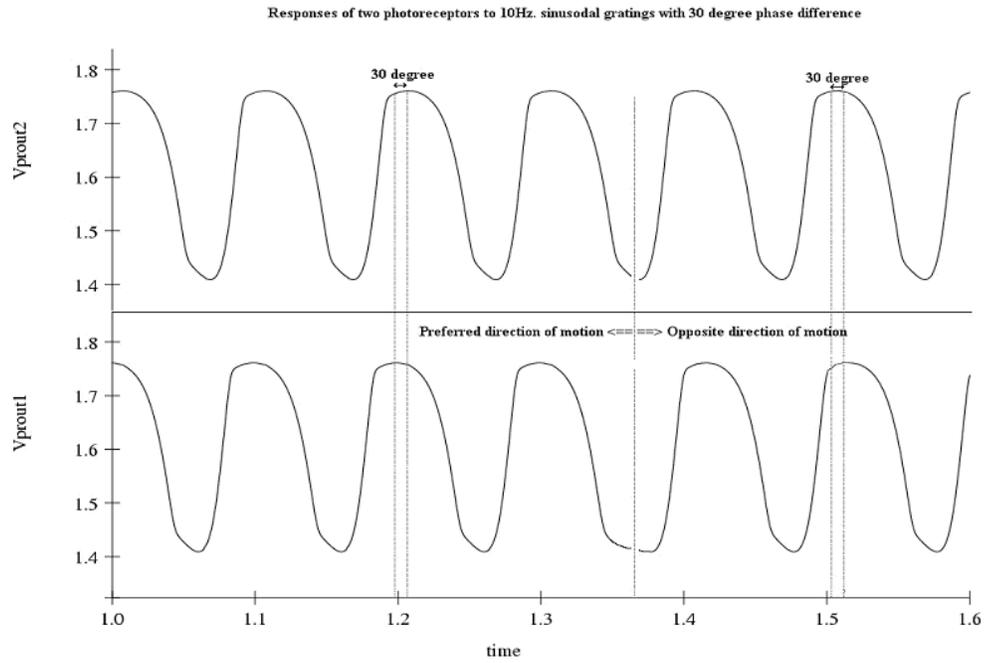
The first version of the circuit is the direct implementation of the algorithm but this circuit can further be simplified by taking only one of the responses of the nonlinear differentiator circuit and using it as an input to the transconductance amplifier (Figure 3.15). In this way, the transistor count can be reduced to 18. The simulation results (Figure 3.16) show that this kind of realization is also enough to obtain the direction of motion. The final output of this circuit can be formulated in terms of differentiator output and photoreceptor voltages as follows:

$$I_{out}(t) = I_{diff2} \tanh\left(\frac{\kappa}{2V_T}(V_{prou} - V_{fb})\right) \quad (3.27)$$

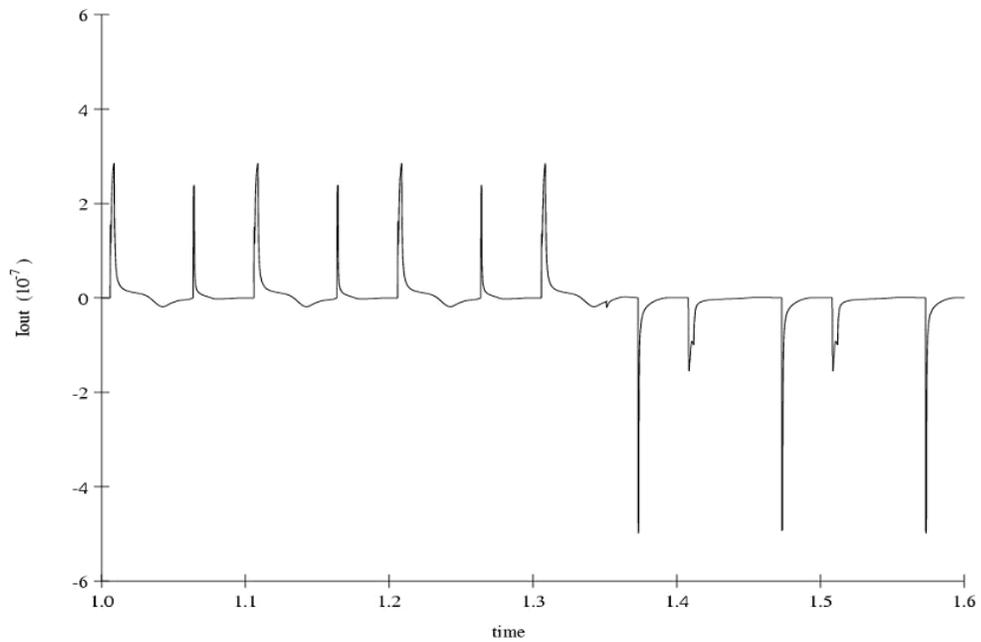
The responses of two implementations are different from the simulation results shown in Figure 3.12, because in the implementations we amplified the difference of two neighboring photoreceptor

outputs instead of using them directly. In addition, we employed nonlinear differentiator and this caused spike-like responses (Figure 3.14b and 3.16). The layouts of these designs are shown in Figures A.4, A.5, A.6 and A.7. Again, the characterization results of this chip are not yet available.





(a)



(b)

Figure 3.14: The response of the first version of the nonlinear differentiator based algorithm. (a) The response of two photoreceptors to a 10Hz sinusoidal grating with 30 degree phase difference. (b) The motion output of the circuit (the direction is changed around 1.37s).



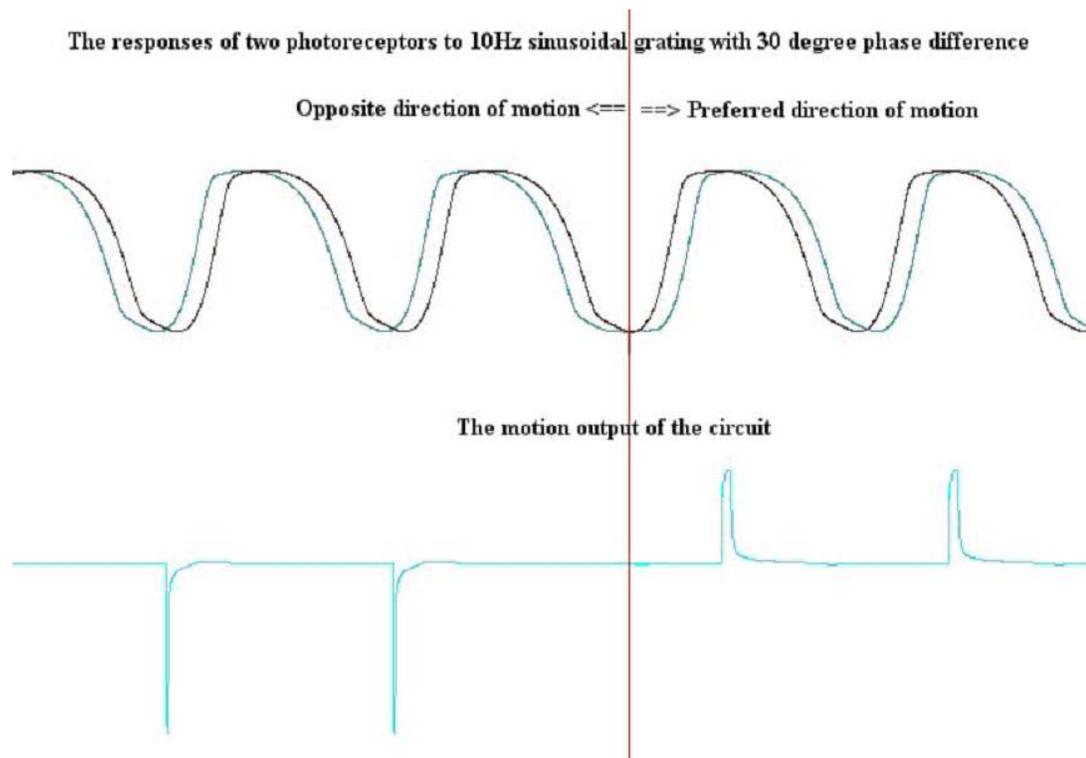


Figure 3.16: The response of the second version of the circuit implementation of the nonlinear differentiator based motion sensor to a 10Hz sinusoidal grating.

## Chapter 4

# Multi-chip implementations of Analog VLSI Motion Sensors

Multi-chip sensors are realized by employing a modular system design approach to increase the flexibility and the computational capabilities of neuromorphic modules. In this kind of implementation, the communication between the modules is achieved by employing an interchip communication protocol. In this study, we utilize an asynchronous interchip communication protocol in designing and implementing different neuromorphic visual models. Each multi-chip system includes one photosensitive sender chip and one motion computing receiver chip to achieve 1-D motion computation. Due to its modular nature, this system can be easily used to compute the 2-D optical flow field by employing one more receiver chip. In addition, the sender chip is designed so that it can be compatible with any intensity based algorithm. The sender transfers only intensity changes instead of the sensed illumination level and is therefore efficient in terms of power consumption.

In this work, we present chip level implementations of three different intensity-based visual motion algorithms, namely, the Adelson-Bergen spatiotemporal energy model [2], the Hassenstein-Reichardt correlation model [31] and the Barlow-Levick motion detection model [6]. These chips have been fabricated in a standard  $1.5\mu\text{m}$  CMOS process and  $2.1 \times 2.1\text{mm}^2$  die size. They have been tested and characterized for different spatiotemporal frequencies and contrast levels. Also, we present the characterization results of these sensors. All the analog motion computations are performed with MOSFETs operating in the subthreshold region to minimize the power consumption of the sensors.

All the sender and receiver chips are composed of P and N pixels that are responsible for positive and negative intensity changes. The high level architecture of these pixels are illustrated in Figure 4.1. This layout is same for all sender and receiver chips.

### 4.1 AER protocol

The Address-Event-Representation (AER) is a very efficient way of solving problems faced in the interchip communication between neuromorphic modules. In this protocol, events are represented with digital pulses and are utilized to transfer temporal changes in the illumination level of the optical image by encoding their analog information. The implementation of this method uses a handshaking protocol and employs two digital control lines and several digital address lines to interface the sender and receiver chips (Figure 4.2a). The functioning of this communication protocol is illustrated in Figure 4.2b. In this protocol, if a request with a valid address is made by the sender then the address lines encode the spatial position of the selected pixel and transfer a binary event to the corresponding pixel in the receiver chip. The acknowledgement signal sent by the receiver leads to falling request and falling acknowledge and the system returns back to its initial condition. In order to assign the interchip communication bus to a particular requesting pixel, an arbitration scheme is utilized in the sender chip. In this way, it is possible to serialize simultaneous events onto a single communication bus. For this purpose we employed a binary tree arbiter that is advantageous since it

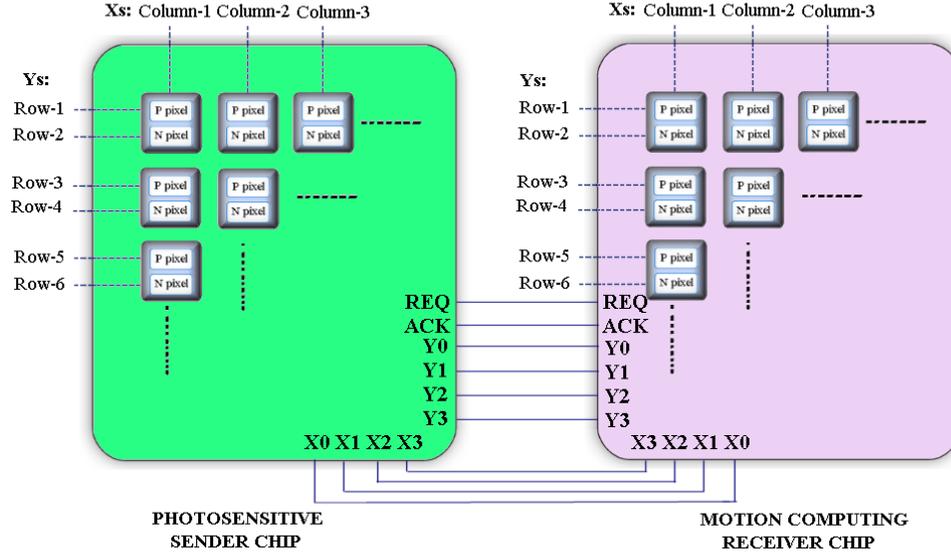


Figure 4.1: The organization of P and N pixels in the sender and receiver chips. Each pixel is composed of P and N pixels that correspond to the negative and positive part of the rectified signal.

can be integrated well with large-array implementations. The details of the hardware implementation of the AER system that we incorporated into our sensors are explained in [45].

## 4.2 Photosensitive sender chip

In this work, we implemented only one type of sender chip that can transfer all the intensity changes in its 2D array of pixels to the corresponding computational pixels in the receiver chip. In this way, the sender chip becomes compatible with different types of receiver chips that can compute motion by making use of changes in the intensity information. The sender pixel shown in Figure 4.4 consists of four circuit building blocks that are described below.

Firstly, we employed an adaptive photoreceptor by Liu [68]. This circuit is shown in Figure 2.2 and described in Section 2.2a. This circuit is used in the sender chip because it adapts to the local light intensity on slow time scales providing high gain for transient signals that are centered on the adaptation point. Moreover, in order to limit the power consumption and increase the efficiency in the operation of sender chip, we used the internal feedback voltage  $V_{fb}$  that represents the adaptation state of the photoreceptor. By using this voltage we successfully removed the offset from the intensity level. The photoreceptor output and its feedback voltage can be related for short and long time scales as explained in Section 2.2.

In the next stage, the response of the photoreceptor is compared with its feedback voltage and converted to a current output by making use of a transconductance amplifier shown in Figure 2.3a. In this way, we obtain a bandpass characteristic from the photoreceptor and transconductance amplifier pair. This characteristic ensures that very high frequencies are attenuated and any offset, or in this case the background level, is removed. The current output of the transconductance amplifier can be represented in terms of the photoreceptor voltages as follows:

$$I_{out} = I_{bias} \tanh\left(\frac{\kappa}{2V_T}(V_{prou} - V_{fb})\right) \quad (4.1)$$

where  $I_{bias}$  is the drain-to-source current that is set by  $V_{diffbias}$  and  $I_{in}$  is the current driven by the

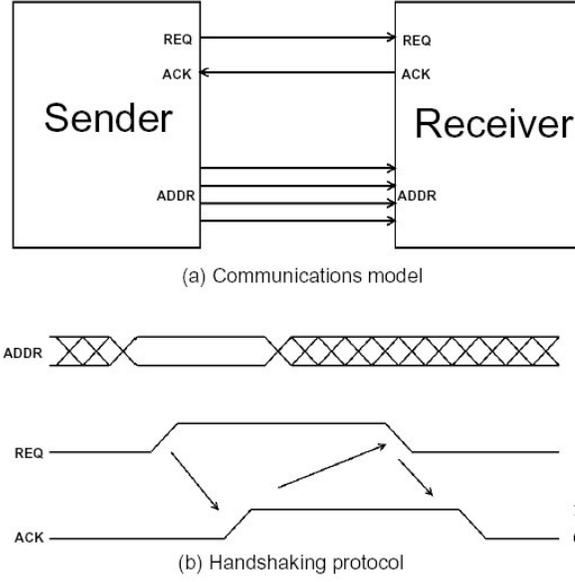


Figure 4.2: AER protocol summary. (a) The model for AER transmission: a sender chip communicates with a receiver chip via request, acknowledge and address lines. (b) The handshaking protocol for transmission using the control and address lines: a request with a valid address leads to an acknowledgment, which in turn leads to falling request and falling acknowledge.

photodiode. For short timescales, the above equation can be written as

$$I_{out} = I_{bias} \tanh\left(\frac{C_2}{2C_1} \ln\left(\frac{I_{in}}{I_0}\right)\right) \quad (4.2)$$

For long timescales since  $V_{prou} = V_{fb}$ ,

$$I_{out} = 0 \quad (4.3)$$

After that, the output current of the transconductance amplifier is rectified by utilizing a full-wave rectifier (Figure 2.4a) in order to acquire separate current representations for negative and positive intensity changes. These currents are compared with predetermined threshold values to set a threshold level over and below the adaptation state of the photoreceptor. In this case, the current outputs of the analog part of the pixel circuitry can be formulated as follows:

$$I_{outpos} = \begin{cases} I_{bias} \tanh\left(\frac{C_2}{2C_1} \ln\left(\frac{I_{in}}{I_0}\right)\right) - I_{leakpos} & \text{when } V_{prou} > V_{fb} \\ 0 & \text{when } V_{prou} < V_{fb} \end{cases}$$

$$I_{outneg} = \begin{cases} |I_{bias} \tanh\left(\frac{C_2}{2C_1} \ln\left(\frac{I_{in}}{I_0}\right)\right)| - I_{leakneg} & \text{when } V_{prou} < V_{fb} \\ 0 & \text{when } V_{prou} > V_{fb} \end{cases}$$

where  $I_{leakpos}$  and  $I_{leakneg}$  are the predetermined leakage currents. The currents  $I_{outpos}$  and  $I_{outneg}$  are sent by the AER communication interface circuit (shown in Figure 4.3) to the corresponding pixels in the receiver chip only if they are bigger than zero. This circuit is a spiking circuit that generates spikes with a frequency proportional to the amplitude level of the current input. The details of the AER interface circuit are explained in [35].

The activation of the AER circuit is limited by the  $V_{leak}$  voltage. In order to request service, the illumination change has to exceed the leakage current created by  $V_{leak}$ . This limitation helps the AER circuit set a minimum signal level that can be transmitted to the receiver chip. When the



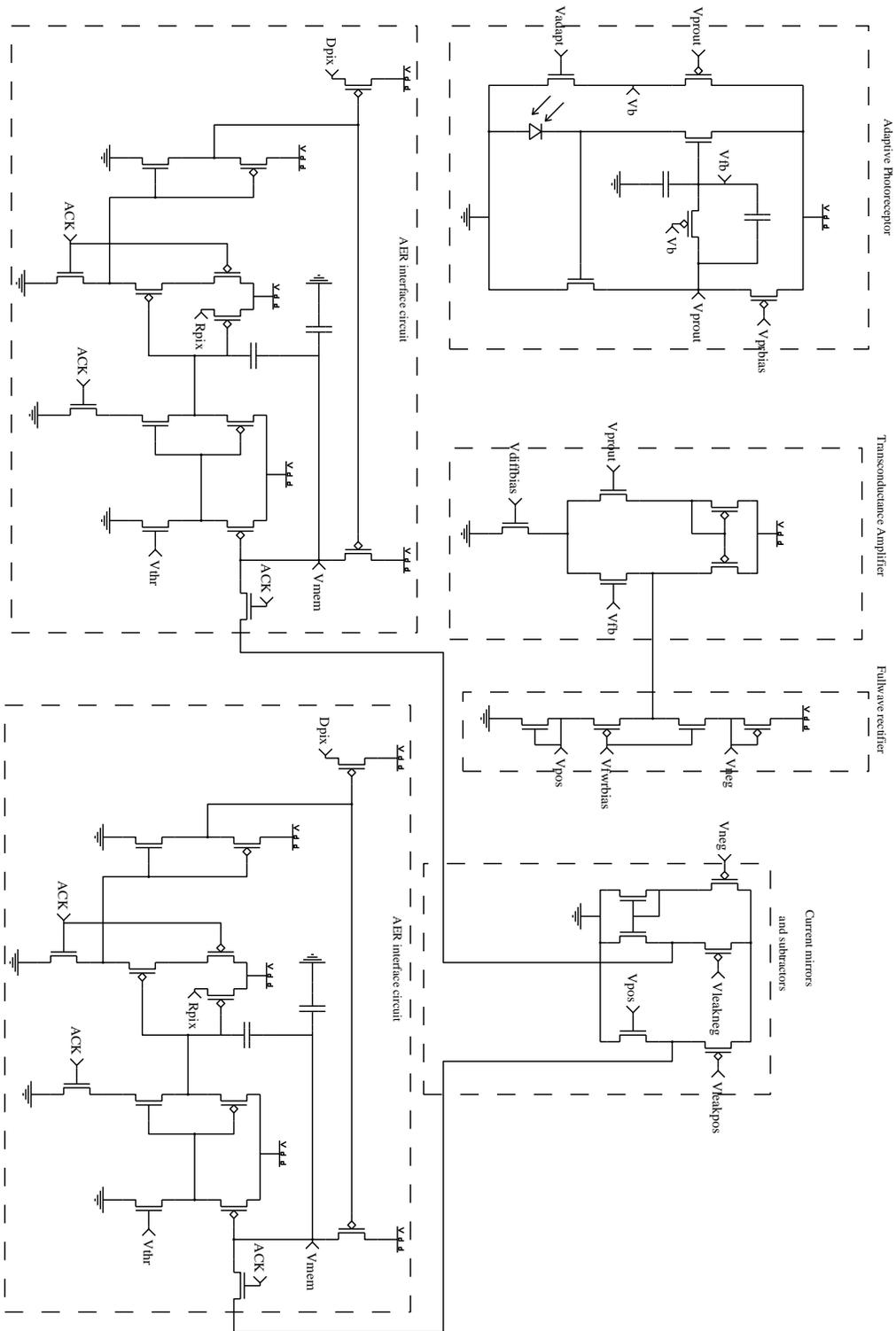


Figure 4.4: Sender chip pixel circuitry.

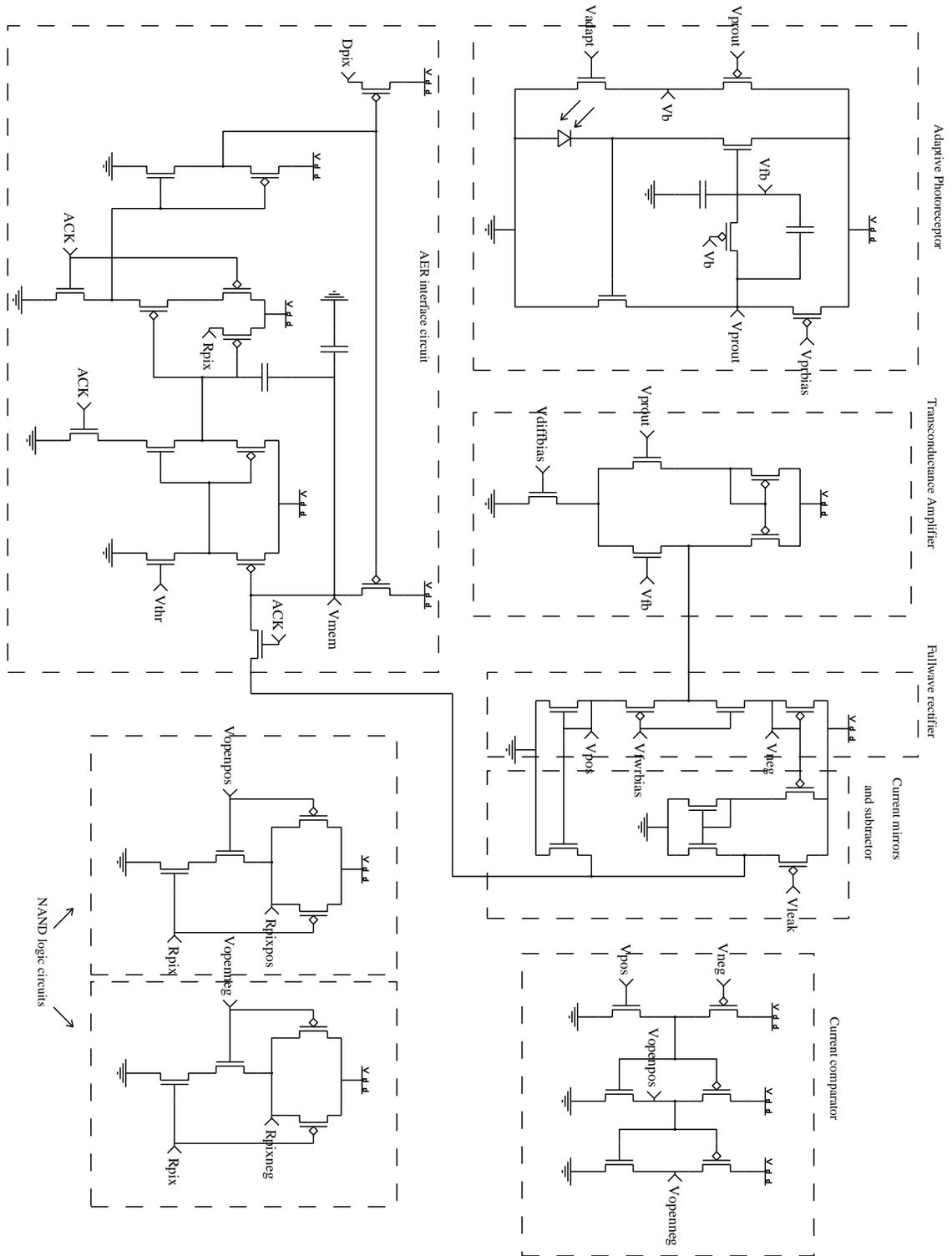


Figure 4.5: The second version of the sender chip pixel circuitry.

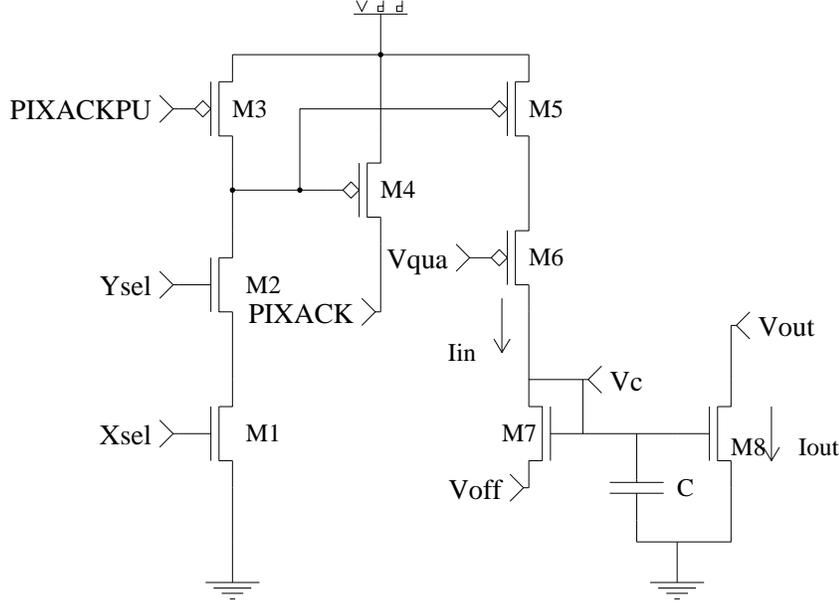


Figure 4.6: AER circuit in the receiver chip.

receiver. Signals are transmitted to the receiver pixels by holding their  $X_{sel}$  and  $Y_{sel}$  signals high (Figure 4.6). This causes  $PIXACK$  to go high which indicates that signal is transmitted. In this way, the AER interface circuit communicates with the peripheral AER system by pulling  $PIXACK$  up for each spike sent.

Whenever  $X_{sel}$  and  $Y_{sel}$  are selected, a charging current through transistor  $M_7$  is created. The amplitude of this current is determined by the bias  $V_{qua}$  and created charges are integrated on the capacitor. In this implementation, the integrating circuit is utilized as a temporal filter. In order to justify that such realization is possible, the output current of the circuit is formulated in terms of  $V_{qua}$ ,  $V_{off}$  and the frequency of spikes coming in.

Firstly, assuming that the current through  $M_7$  is  $I_{M_7}$ , the current through the capacitor is  $I_c$  and the input current is  $I_{in}$  (which is created by the transistor  $M_6$  whenever  $X_{sel}$  and  $Y_{sel}$  are selected), then the relation between these currents can be shown as follows:

$$I_{in} = I_{M_7} + I_c \quad (4.4)$$

In the calculations, we presume that  $I_{in}$  is constant and exist whenever AER interface circuit is activated. The capacitor current can be written as  $I_c = C \frac{\delta V_c}{\delta t}$ . Since the current  $I_{M_7}$  is amplified by the voltage gain  $V_{off}$ , the output current can be formulated in terms of  $I_{M_7}$  and  $V_{off}$  as  $I_{out} = I_{M_7} e^{\frac{V_{off}}{V_T}}$ . In order to represent  $I_{out}$  in terms of  $I_{in}$ , we use the chain rule to relate the time derivative of  $I_{out}$  to the time derivative of  $V_c$ .

$$\frac{\delta I_{out}}{\delta t} = \frac{\delta I_{out}}{\delta V_c} \frac{\delta V_c}{\delta t} \quad (4.5)$$

Since  $I_{out} = I_o e^{\frac{\kappa V_c}{V_T}}$ ,

$$\frac{\delta I_{out}}{\delta V_c} = \frac{\kappa I_{out}}{V_T} \quad (4.6)$$

and since  $\frac{\delta V_c}{\delta t} = \frac{I_c}{C}$  and  $I_c = I_{in} - I_{M_7}$ , by writing  $I_{M_7}$  in terms of  $I_{out}$  we can obtain the following

equation:

$$\frac{\delta V_c}{\delta t} = \frac{1}{C}(I_{in} - I_{out}e^{-\frac{V_{off}}{V_T}}) \quad (4.7)$$

By combining Equations 4.7 and 4.6, we can show that the relation between  $I_{out}$  and  $I_{in}$  can be formulated as follows:

$$\frac{\delta I_{out}}{\delta t} = \frac{\kappa I_{out}}{V_T} \frac{1}{C}(I_{in} - I_{out}e^{-\frac{V_{off}}{V_T}}) = \frac{1}{K}I_{out}(I_{in} - \alpha I_{out}) \quad (4.8)$$

where  $K = \frac{CV_T}{\kappa}$  and  $\alpha = e^{-\frac{V_{off}}{V_T}}$ . This equation can be solved by integrating it as follows:

$$\int_{I_{out}(0)}^{I_{out}(t)} \frac{\delta I_{out}}{I_{out}(I_{in} - \alpha I_{out})} = \frac{1}{K} \int_0^t \delta t \quad (4.9)$$

$$\frac{1}{I_{in}} \int_{I_{out}(0)}^{I_{out}(t)} \left( \frac{1}{I_{out}} + \frac{\alpha}{(I_{in} - \alpha I_{out})} \right) \delta I_{out} = \frac{t}{K} \quad (4.10)$$

$$\ln\left(\frac{I_{out}(t)}{I_{in} - \alpha I_{out}(t)} \frac{I_{in} - \alpha I_{out}(0)}{I_{out}(0)}\right) = \frac{I_{in} t}{K} \quad (4.11)$$

From the above equation we can solve and find  $I_{out}$ ,

$$I_{out}(t) = \frac{I_{in}}{2\alpha} \frac{1 - \frac{I_{in} - \alpha I_{out}(0)}{\alpha I_{out}(0)} e^{-\frac{I_{in} t}{K}}}{1 + \frac{I_{in} - \alpha I_{out}(0)}{\alpha I_{out}(0)} e^{-\frac{I_{in} t}{K}}} + \frac{I_{in}}{2\alpha} \quad (4.12)$$

The above equation can be simplified further as follows:

$$I_{out}(t) = \begin{cases} \frac{I_{in}}{2\alpha} \tanh\left(\frac{I_{in}}{2K}t - \frac{1}{2}\ln\left(\frac{I_{in} - \alpha I_{out}(0)}{\alpha I_{out}(0)}\right)\right) + \frac{I_{in}}{2\alpha} & \text{if } I_{in} > \alpha I_{out}(0) \\ \frac{I_{in}}{2\alpha} \coth\left(\frac{I_{in}}{2K}t - \frac{1}{2}\ln\left(\frac{I_{in} - \alpha I_{out}(0)}{\alpha I_{out}(0)}\right)\right) + \frac{I_{in}}{2\alpha} & \text{if } I_{in} < \alpha I_{out}(0) \end{cases}$$

The above equation is valid when  $I_{in}$  is zero or no more spikes received. If AER interface circuit is not activated then  $I_{in}$  becomes zero. In this case,

$$\frac{\delta I_{out}}{\delta t} = \frac{1}{K}I_{out}(I_{in} - \alpha I_{out}) = -\frac{\alpha}{K}I_{out}^2 \quad (4.13)$$

$$\frac{\delta I_{out}}{I_{out}^2} = -\frac{\alpha}{K}\delta t \quad (4.14)$$

Integrating the above equation,

$$-\int_{I_{out}(0)}^{I_{out}(t)} \frac{\delta I_{out}}{I_{out}^2} = \frac{\alpha}{K} \int_0^t \delta t \quad (4.15)$$

$$I_{out}(t) = \frac{I_{out}(0)}{\frac{\alpha I_{out}(0)}{K}t + 1} \quad (4.16)$$

By employing this equation in the calculations, the response of the circuit to a pulse train can be found. Assuming the period of the pulse train is T and the charge integrated in the capacitor

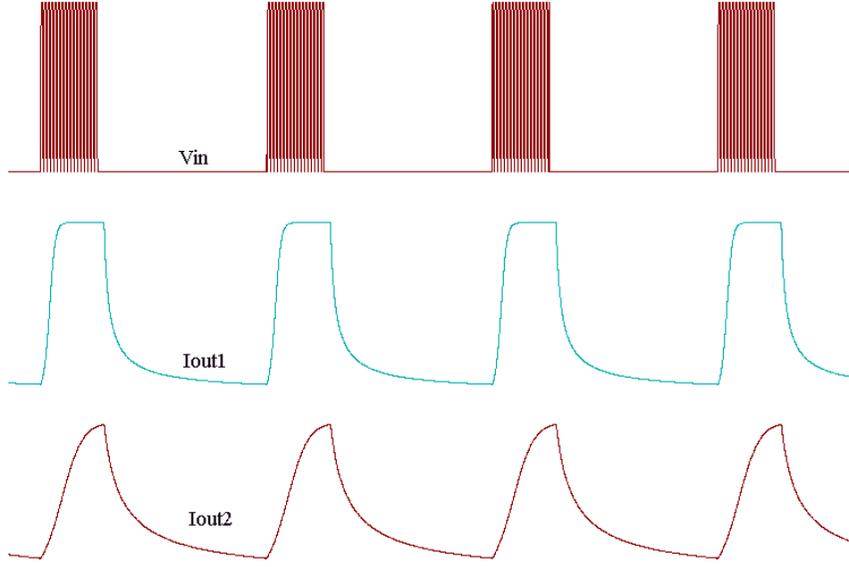


Figure 4.7: The current outputs of two integrators biased with different  $V_{qua}$  and  $V_{off}$  biases. A periodic spike train,  $V_{in}$  is provided to two integrators.  $V_{qua1}$  is made smaller than  $V_{qua2}$  (relative to  $V_{dd}$ ) and this yields  $I_{out2}$  to be the filtered version of  $I_{out1}$ . The difference in the amplitude of these currents is compensated by biasing  $V_{off2}$  larger than  $V_{off1}$ .

per pulse is  $Q_{cap}$ , then for the time interval  $t_0 + nT \leq t < t_0 + (n+1)T$ ,

$$I_{out}(t) = \frac{I_{out}(t_0 + nT)}{\frac{\alpha I_{out}(t_0 + nT)}{K}(t - (t_0 + nT)) + 1} \quad (4.17)$$

This equation corresponds to the decay phase of the signal. When it is updated by another pulse, the output current becomes

$$I_{out}(t_0 + nT) = \frac{1}{\frac{1}{I_T} + \left(\frac{1}{I_{out}(t_0)} - \frac{1}{I_T}\right)(1 + \beta)^{-n}} \quad (4.18)$$

where  $\beta = e^{\frac{Q_{cap}}{K}} - 1$  and  $I_T = \beta \frac{K}{\alpha T}$ . The peak response becomes  $I_T$ , when  $(1 + \beta)^{-n} \ll 1$ . When the current  $I_{in}$  is switched off, the output current converges to  $I = \frac{I_T}{1 + \beta}$ .

This circuit is employed to integrate the positive and negative part of the intensity signal sent by the sender chip and to obtain a similar configuration as has been achieved with monolithic implementations. As an example consider the case when the positive part of the signal sent by the sender is integrated by two integrators that have different  $V_{qua}$  and  $V_{off}$  biases to obtain a delay between these two integrated signals. Assuming that the first integrator has  $V_{qua1}$  and  $V_{off1}$  biases and the second integrator has  $V_{qua2}$  and  $V_{off2}$  biases, when  $(1 + \beta)^{-n} \ll 1$ , the peak responses of the integrated signals become

$$I_{out1} = I_{T1} = \left(e^{\frac{\kappa Q_{cap1}}{CV_T}} - 1\right) \frac{CV_T}{\kappa T} e^{\frac{V_{off1}}{V_T}} \quad (4.19)$$

$$I_{out2} = I_{T2} = \left(e^{\frac{\kappa Q_{cap2}}{CV_T}} - 1\right) \frac{CV_T}{\kappa T} e^{\frac{V_{off2}}{V_T}} \quad (4.20)$$

Their decaying responses converge to the current levels which can be shown as follows:

$$I_{out1} = \frac{I_{T1}}{1 + \beta} = (1 - e^{-\frac{\kappa Q_{cap1}}{CV_T}}) \frac{CV_T}{\kappa T} e^{\frac{V_{off1}}{V_T}} \quad (4.21)$$

$$I_{out2} = \frac{I_{T2}}{1 + \beta} = (1 - e^{-\frac{\kappa Q_{cap2}}{CV_T}}) \frac{CV_T}{\kappa T} e^{\frac{V_{off2}}{V_T}} \quad (4.22)$$

where  $Q_{cap}$  is mainly controlled by  $V_{qua}$ . The amplitude levels of the output signals can be changed by using both  $V_{off}$  and  $V_{qua}$ . Besides, the characteristic of the rising edges of the signals are determined by the expression  $(e^{\frac{\kappa Q_{cap}}{CV_T}} - 1)$  and the decaying parts are governed by the expression  $(1 - e^{-\frac{\kappa Q_{cap}}{CV_T}})$ . These biases help us obtain a low-pass characteristic from the integrator circuit. In Figure 4.7, the incoming spikes,  $V_{in}$ , are integrated by two integrators biased with different  $V_{qua}$  and  $V_{off}$  biases. The current outputs of these integrators are defined as  $I_{out1}$  and  $I_{out2}$ , and the difference between their behaviors are illustrated.  $I_{out2}$  can be used as a low-pass filtered version of  $I_{out1}$ . When  $V_{qua1}$  is made smaller than  $V_{qua2}$  relative to  $V_{dd}$ ,  $I_{out2}$  is delayed more than  $I_{out1}$ . The difference in the amplitude of these currents is compensated by biasing  $V_{off2}$  larger than  $V_{off1}$ .

After the integration stage, motion computation algorithms are implemented in the receiver chips. In this work, we present three intensity based models that utilize the integration stage of the AER circuitry as a temporal filter.

### 4.3.1 Hassenstein-Reichardt model

Hassenstein and Reichardt deduced a model for motion detection on the basis of the optomotor responses obtained from the beetle *Chlorophanus* [31]. They succeeded in eliciting a measurable optomotor response by presenting the animal with sequences of light and dark steps that simulated motion in a given direction. After analyzing the responses to these sequences, it has been concluded that motion detection by the nervous system requires an interaction of signals from two directly neighboring or next neighboring ommatidia.

The Hassenstein and Reichardt model is a correlation-type motion detector that operates directly on filtered versions of the retinal light intensity distribution and assumes a multiplication for the interaction of its two input channels as illustrated in Figure 4.8. When a detector receives a signal in the input channel which is activated first by a moving stimulus, it is delayed by an appropriate time interval  $\Delta t$ , and then the signals in both input channels tend to coincide at the multiplication stage resulting in a large response amplitude. Conversely, when the temporal sequence of stimulation is reversed (corresponding to motion in the null direction), the separation of both signals is further increased by the detector delay, resulting in only small responses.

The correlation scheme includes the delay and multiplication stages to eliminate the response components that are not specifically a result of stimulus motion but are induced by correlated input signals such as background luminance. The outputs of multiplication stages are then subtracted leading to responses of the same amplitude but of different signs for motion in opposite directions.

While a great deal of experimental evidence supports the Reichardt correlator as a mechanism in biological motion detection, the correlator does not signal the true image velocity [62]. However, the predictable statistics of natural images imply a consistent correspondence between mean correlator response and velocity, enabling the Reichardt correlator to act as a practical velocity estimator. In addition, analysis and simulations [19] suggest that processes commonly found in visual systems, such as pre-filtering, response compression, integration and adaptation improve the reliability of velocity estimation and expand the range of velocities coded by the model.

In a variety of studies the Reichardt correlation model has been implemented on analog VLSI chips to estimate the optical motion field created by an image motion [4], [30] [66], [67]. Also, Liu [55] and Harrison [30] used this model to mimic the fly's motion detection system. In this thesis, we present an implementation of the Reichardt model by incorporating the modular strategy.

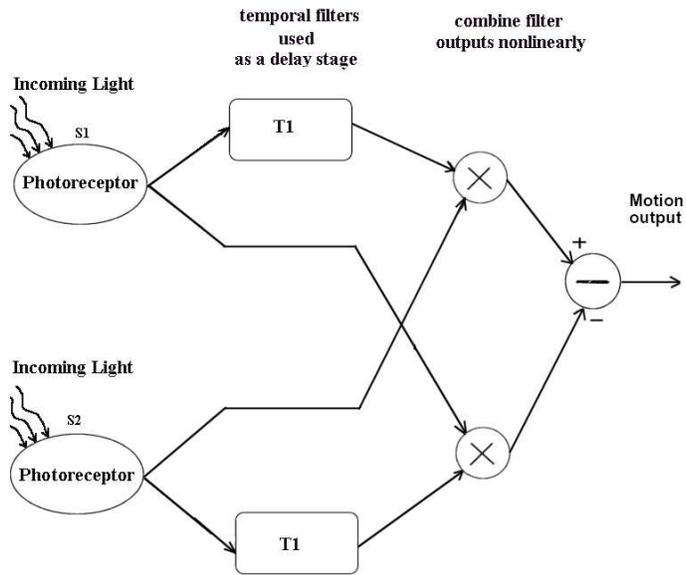


Figure 4.8: The Reichardt model.

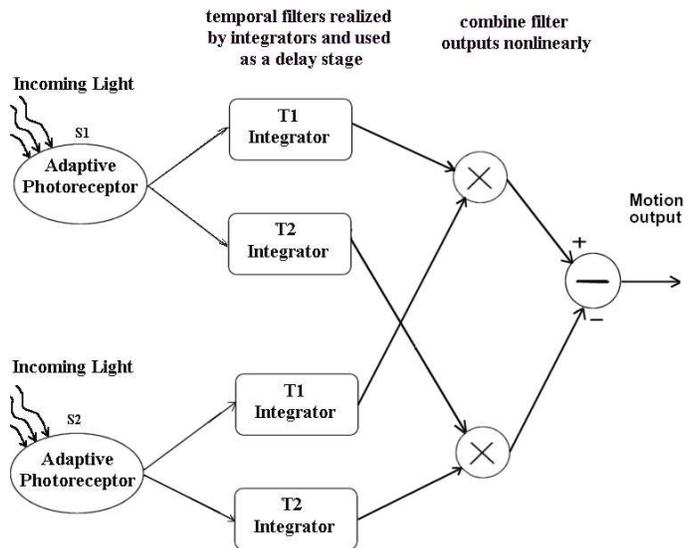


Figure 4.9: Modified version of the Reichardt model.

#### 4.3.1.1 Modified form:

In order to achieve a correlation-based motion computation and implement the Reichardt model in analog VLSI hardware by employing the multi-chip strategy, we modified the canonical algorithm (shown in Figure 4.8). The modified model is shown in Figure 4.9.

In order to realize the algorithm, we have to remove the DC component of the illumination level. In this implementation, this was achieved by taking the difference between the photoreceptor output and its adaptation state by employing a transconductance amplifier. The Reichardt model is a correlation based algorithm, so the maximum response from the model can be retrieved when the phase difference between the neighboring photoreceptors is close to 90 degrees. In the original model this is obtained by using delay lines. In the modified model, we employed integrators to be able to introduce different amounts of delay to the signals. Finally, the nonlinearity stage is realized by using the multiplication operation as is the case in the original form of the model.

#### 4.3.1.2 Expected response:

In order to predict the characteristics of the Hassenstein-Reichardt model, we use its response to a sinusoidal grating. Assume that the spatial pattern observed by the photoreceptors is a sinusoid with maximum amplitude  $A$ , contrast  $C$  and spatial frequency  $f_s$  cycles/degree, travelling with velocity  $v$ , then the luminance pattern can be represented as a function of time and space as follows:

$$I(x, y, t) = A(1 + C \cdot \sin(2\pi f_s x + 2\pi f_t t)) \quad (4.23)$$

where  $f_t = v \cdot f_s$ . Accordingly, the inputs of the correlators can be defined as follows:

$$I_1(t) = A(1 + C \cdot \sin(2\pi f_t t + 2\pi f_s x)) \quad (4.24)$$

$$I_2(t) = A(1 + C \cdot \sin(2\pi f_t t + 2\pi f_s x \pm 2\pi f_s \Delta x)) \quad (4.25)$$

where  $\Delta x$  corresponds to the distance between neighboring photoreceptors, and plus and minus indicate the direction of motion. Assuming that the temporal filter has a frequency response  $H(f_t) = F(f_t) \cdot e^{i\Theta(f_t)}$ , where  $F(f_t)$  and  $\Theta(f_t)$  are real-valued functions indicating amplitude and phase, then after removing the mean of the signals  $I_1$  and  $I_2$  we can obtain the delayed signals,

$$I_{D1}(t) = A \cdot F(f_t) \cdot C \cdot \sin(2\pi f_t t + 2\pi f_s x + \Theta(f_t)) \quad (4.26)$$

$$I_{D2}(t) = A \cdot F(f_t) \cdot C \cdot \sin(2\pi f_t t + 2\pi f_s x \pm 2\pi f_s \Delta x + \Theta(f_t)) \quad (4.27)$$

By applying the computation performed in the model and simplifying with the trigonometric identities  $\cos(\alpha) \cdot \cos(\theta) = \frac{1}{2}[\cos(\alpha + \theta) + \cos(\alpha - \theta)]$  and  $\cos(\alpha + \theta) = \cos(\alpha) \cdot \cos(\theta) - \sin(\alpha) \cdot \sin(\theta)$ , we can find the detector output as follows:

$$R(t) = I_{D1} \cdot I_2 - I_{D2} \cdot I_1 \quad (4.28)$$

$$= \frac{C^2 \cdot F(f_t)}{2} \cdot (\sin(\Theta(f_t) \mp 2\pi f_s \Delta x) - \sin(\Theta(f_t) \pm 2\pi f_s \Delta x)) \quad (4.29)$$

$$= C^2 \cdot F(f_t) \cdot \sin(\Theta(f_t)) \cdot \sin(\pm 2\pi f_s \Delta x) \quad (4.30)$$

Further assuming that the temporal filter is a first-order low-pass filter with impulse response

$d(t) = \frac{1}{\tau}e^{-\frac{t}{\tau}}$  for  $t > 0$ , the frequency response of this filter can be found as the following expression:

$$D(f_t) = \frac{1}{\sqrt{1 + (2\pi f_t \tau)^2}} e^{-i \arctan(2\pi f_t \tau)} \quad (4.31)$$

From this equation, we can find  $F(f_t)$  and  $\Theta(f_t)$ ,

$$F(f_t) = \frac{1}{\sqrt{1 + (2\pi f_t \tau)^2}} \quad (4.32)$$

$$\Theta(f_t) = \arctan(2\pi f_t \tau) \quad (4.33)$$

$$\sin(\Theta(f_t)) = \frac{2\pi f_t \tau}{\sqrt{1 + (2\pi f_t \tau)^2}} \quad (4.34)$$

Finally, we can obtain the motion output by using above equations.

$$R(t) = \frac{C^2}{2\phi\tau} \cdot \frac{f_t}{(f_t)^2 + 1/(2\phi\tau)^2} \cdot \sin(\pm 2\pi f_s \Delta x) \quad (4.35)$$

In 1D motion computation,  $f_x = f_s \sin(\theta)$  where  $\theta$  is the orientation of the stimulus, and the motion output becomes

$$R(t) = \frac{C^2}{2\phi\tau} \cdot \frac{f_t}{(f_t)^2 + 1/(2\phi\tau)^2} \cdot \sin(\pm 2\pi f_x \sin(\theta) \Delta x) \quad (4.36)$$

From the final formula, we observe that the model yields a square-law response to contrast variations and a sinusoidal response to stimulus orientation. The spatial frequency response of the model is determined the term  $\sin(2\pi f_x \sin(\theta) \Delta x)$ . Accordingly, the peak of the response occurs when  $2\pi f_x \sin(\theta) \Delta x = \frac{\pi}{2}$ . Finally, the temporal response of the model is determined by the frequency response of the low-pass filter since the bandwidth of the photoreceptor is assumed to be wider compared to the temporal filter. Therefore, the temporal frequency peak in the final output can be altered by changing the tuned temporal frequency of the low-pass filter.

#### 4.3.1.3 Multi-chip implementation of the model:

The multi-chip implementation of the model is realized with sender and receiver chips. The main aim in utilizing the modular strategy in motion processing is to decrease the computational load of the front-end chip and increase the computational capacity of the overall system. Therefore, the computation performed in the front-end chip is minimized by splitting the motion computation into sender and receiver chips.

The details of the sender chip are explained in Section 4.2. Basically, the function of the sender chip in this implementation is to sense the illumination changes in the environment and send this information to the receiver chip.

The receiver part of the multi-chip system is composed of the serial scanners, communication interface and corresponding pixels of the sender chip. Each pixel in the receiver includes P and N pixels as shown in Figure 4.11a and these pixels correspond to the positive and negative parts of the rectified signal in the sender chip. These sub-pixels include an interface to integrate the incoming spikes. The voltage level of these integrated spikes is used to perform motion computation in the P and N pixels. The final computation is performed by the interactions between the P and N parts of two neighboring pixels.

As seen in the P-pixel (Figure 4.11), the same incoming spikes are integrated by two integrator circuits. These integrators are not identical in terms of the delay they provide; one of them can be

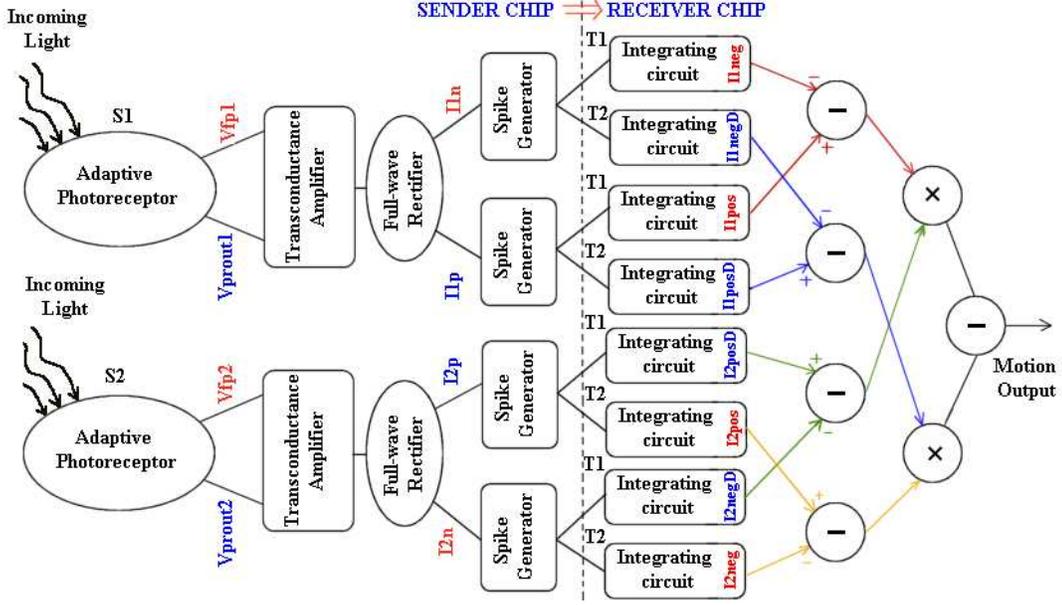


Figure 4.10: A multi-chip implementation of the Reichardt model.

used to obtain an extra delay that is needed to perform the correlation. In this way, one can obtain the delayed signal without using an extra low-pass circuit.

The nonlinearity needed in motion computation is achieved by using four-quadrant current multiplier circuit [13] (Figure 4.11b). After the subtraction stage, the final output that represents the extracted motion information can be obtained. The final output of the circuit can be formulated as follows:

$$\begin{aligned}
 I_{out} = & I_{1posD} \cdot I_{2pos} + I_{1negD} \cdot I_{2neg} - I_{2posD} \cdot I_{1pos} - I_{2negD} \cdot I_{1neg} \\
 & - I_{1posD} \cdot I_{2neg} - I_{1negD} \cdot I_{2pos} + I_{2posD} \cdot I_{1neg} + I_{2negD} \cdot I_{1pos}
 \end{aligned} \quad (4.37)$$

This equation corresponds to a familiar output expression of the correlation scheme realized by the original Reichardt model.

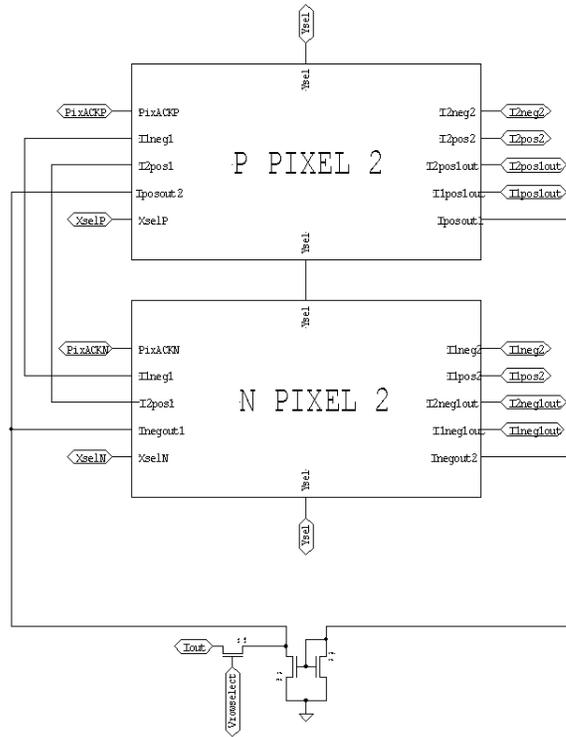
$$I_{out} = I_{1D} \cdot I_2 - I_{2D} \cdot I_1 \quad (4.38)$$

The final motion output of the sensor gives an indication whether the motion is in the preferred or null direction. The circuit implementation of the model is achieved with 65 transistors and 4 capacitors in the P and N pixels of the receiver chip. The layout of the receiver pixel and chip are shown in Figures A.10 and A.11, respectively. The Reichardt receiver includes  $6 \times 7$  pixels.

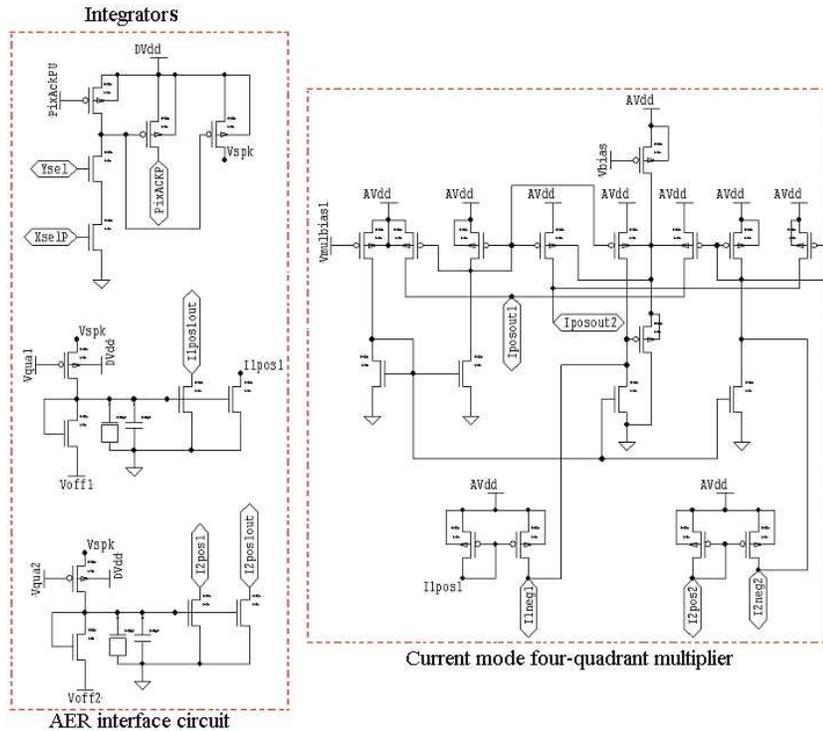
### 4.3.2 Adelson-Bergen spatiotemporal energy model

The Adelson-Bergen algorithm is a spatiotemporal algorithm which obtains its direction selectivity by integrating quadrature filters with a nonlinearity function. This algorithm extracts the Fourier energy in a band of spatiotemporal frequencies regardless of the phase of the stimulus. It is used to explain the visual pathways of primate cortical complex cells [61] and it has been shown that this algorithm corresponds to the elaborated Reichardt correlation model [65].

In order to perceive the motion in continuous or sampled displays, there must be energy in the signal that has appropriate spatiotemporal orientation. This spatiotemporal orientation can be detected by using two units that act as linear spatiotemporal filters (the one at the left-hand side of Figure 4.12 has cosine (even) phase, whereas that at the right-hand side has sine (odd) phase). Then squaring and summing the two unit outputs, a measure of local motion energy can be acquired.



(a)



(b)

Figure 4.11: The Reichardt receiver chip pixel circuitry. (a) The structure of the main pixel in the Reichardt receiver chip. (b) The P pixel circuitry in the Reichardt receiver chip, the circuitry is same in the N pixel.

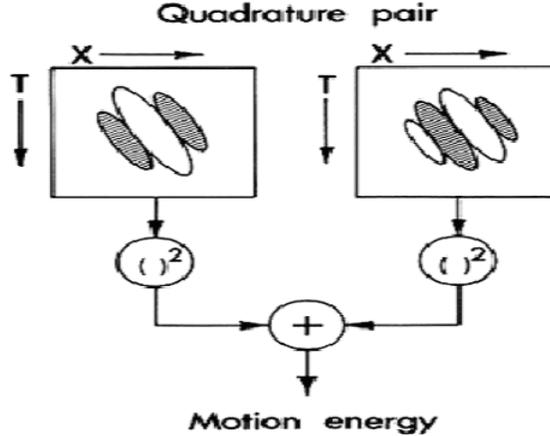


Figure 4.12: The structure of the Adelson-Bergen model in terms of linear filters. Two linear filters, whose responses are 90 degrees out of phase, form a quadrature pair. If their responses are squared and summed, the resulting signal gives a phase-independent measure of local motion energy. Reproduced from [2] without permission.

In this model, the two Gabor functions are sine and cosine functions weighted by the same Gaussian window and they are utilized to extract the energy within a spatiotemporal-frequency band. The resulting response will always be positive, and it will grow and fall smoothly in the region of the moving edge. As a result, the response will be sensitive to the direction of motion but insensitive to the sign of the stimulus contrast [2].

A software version of the Adelson-Bergen algorithm was implemented on a general-purpose analog neural computer by Etienne-Cummings [25]. Later, Higgins and Korropati [37] implemented an analog VLSI sensor based on this algorithm. In this study, we show that the multi-chip idea can be successfully utilized to realize the Adelson-Bergen algorithm without using explicit temporal filters in the implementation.

#### 4.3.2.1 Canonical Form:

The canonical form of the Adelson-Bergen algorithm is shown in Figure 4.13. This scheme is used to perform 1D motion detection by employing quadrature spatial and temporal filters. In addition to the spatial and temporal filtering, a certain degree of nonlinearity is introduced to the signal to obtain the direction selectivity and nonseparable motion energy. In the original form of the model, it is proposed that the spatial filters can be realized by making use of the quadrature Gabor filters. These filters can be mathematically represented as follows:

$$f_{s1}(x) = e^{-\frac{x^2}{2\sigma^2}} \cos(w_x x) \quad (4.39)$$

$$f_{s2}(x) = e^{-\frac{x^2}{2\sigma^2}} \sin(w_x x) \quad (4.40)$$

Moreover, the temporal filters are suggested to be second and third derivatives of the Gaussian function.

$$f_{t1}(t) = (kt)^3 e^{-kt^2} \left[ \frac{1}{3!} - \frac{(kt)^2}{(3+2)!} \right] \quad (4.41)$$

$$f_{t2}(t) = (kt)^5 e^{-kt^2} \left[ \frac{1}{5!} - \frac{(kt)^2}{(5+2)!} \right] \quad (4.42)$$

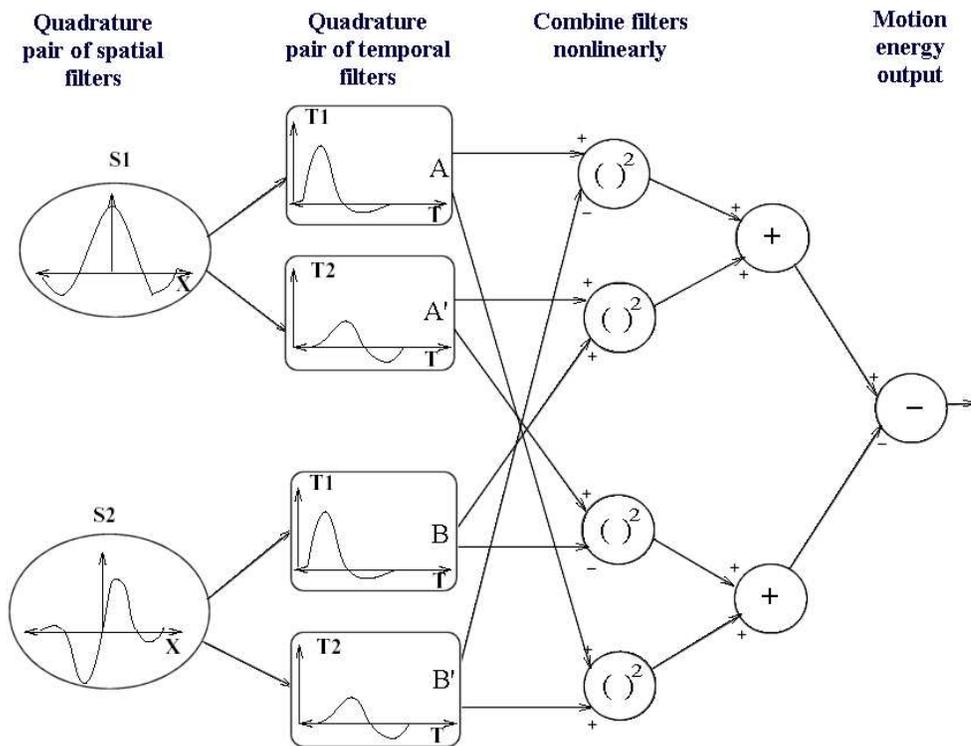


Figure 4.13: The original form of the Adelson-Bergen model. The same spatial and temporal filters are used. Sums and differences generate directionally selective filters. Sums of squares of quadrature pairs give motion energy for each direction. The difference between the rightward and leftward signals gives the final output. Reproduced from [2] without permission.

In case of a one-dimensional sinusoidal grating with amplitude  $I$ , spatial frequency  $w_s$ , and temporal frequency  $w_t$ , the intensity change can be represented as follows:

$$I(x, y, t) = I \cdot \sin(w_t t + w_s x) \quad (4.43)$$

By performing the computations illustrated in Figure 4.13, we can find the final motion energy output. Firstly, from the spatial filters we obtain the following expressions:

$$f_{left}(x, t) = I \cdot |f_{s1}| \cdot \sin(w_t t + w_s x + \phi_{s1}(w_s)) \quad (4.44)$$

$$f_{right}(x, t) = I \cdot |f_{s2}| \cdot \sin(w_t t + w_s x + \phi_{s2}(w_s)) \quad (4.45)$$

where  $|f_{s1}|$  and  $|f_{s2}|$  are the magnitudes of the spatial filters and  $\phi_{s1}(w_s)$  and  $\phi_{s2}(w_s)$  are their phases. After introducing two temporal filters with magnitudes  $|h_{t1}|$  and  $|h_{t2}|$  and phases  $\phi_{t1}$  and  $\phi_{t2}$ , we can obtain the following equations:

$$A(x, t) = I|f_{s1}||f_{t1}| \sin(w_t t + w_s x + \phi_{s1}(w_s) + \phi_{t1}(w_t)) \quad (4.46)$$

$$A'(x, t) = I|f_{s1}||f_{t2}| \sin(w_t t + w_s x + \phi_{s1}(w_s) + \phi_{t2}(w_t)) \quad (4.47)$$

$$B(x, t) = I|f_{s2}||f_{t1}| \sin(w_t t + w_s x + \phi_{s2}(w_s) + \phi_{t1}(w_t)) \quad (4.48)$$

$$B'(x, t) = I|f_{s2}||f_{t2}| \sin(w_t t + w_s x + \phi_{s2}(w_s) + \phi_{t2}(w_t)) \quad (4.49)$$

As it is illustrated in the Figure 4.13, the final motion energy output is  $O = 4(AB' - A'B)$ , and using trigonometric identities, we can obtain the final expression as follows:

$$O = 4I^2|f_{s1}||f_{s2}| \sin(\phi_{s1} - \phi_{s2})|f_{t1}||f_{t2}| \sin(\phi_{t1} - \phi_{t2}) \quad (4.50)$$

This algorithm theoretically yields very reliable results in motion computation. However, in order to realize the algorithm in hardware, the algorithm should be modified adequately.

#### 4.3.2.2 Modified Form:

In this study, the Adelson-Bergen algorithm has been implemented (in a modified form; see Figure 4.14) by making use of neuromorphic principles [58] and simplified without modifying the basic idea of the spatiotemporal energy model.

Firstly, the spatial filtering in the model is trivialized by simply using photoreceptor outputs separated by a  $\Delta x$  spatial distance between adjacent photoreceptors. Secondly, temporal filters in the model are implemented by employing integrating circuits. In this thesis, we have demonstrated that integrating receiver circuitry can be used to attain phase differences. This novel technique of using the integrator as a temporal filter enables us to exploit the advantages of multi-chip strategy in motion computation and to decrease the computational overload. Finally, the nonlinearity required to realize the algorithm is attained in the implementation by making use of the mathematical properties of rectification. In this study we show that absolute value operation can be utilized instead of multipliers to obtain the nonlinearity needed for motion computation.

#### 4.3.2.3 Expected Response:

When the Adelson-Bergen motion sensor is exposed to a one-dimensional sinusoidal grating with amplitude  $A$ , contrast  $C$ , spatial frequency  $f_s$ , orientation  $\theta$  relative to the sensor's preferred orien-

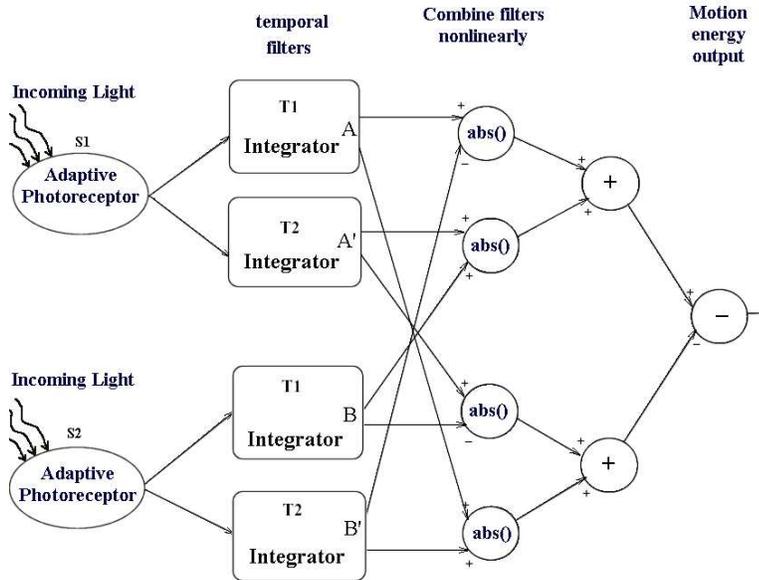


Figure 4.14: Modified version of the Adelson-Bergen spatiotemporal energy model.

tation, and temporal frequency  $f_t$ ,

$$I(x, y, t) = A \cdot (1 + C \cdot \sin(2\pi f_t t + 2\pi f_s \cdot (x \cos(\theta) + y \sin(\theta)))) \quad (4.51)$$

then in case of using a squaring operation (instead of employing the absolute value operation) and a low-pass filter to obtain a closed-form solution, the mean output of the sensor with neighboring photoreceptors separated with  $\delta$ , can be computed as follows:

$$A^2 \cdot P(f_t)^2 \cdot C^2 \cdot \sin(2\pi f_s \delta \cos(\theta)) \cdot H(f_t) \cdot \sin(\phi_t(f_t)) \quad (4.52)$$

where  $P(f_t)$  is the magnitude of the photoreceptor temporal frequency response, and  $H(f_t)$  and  $\phi_t(f_t)$  are respectively the magnitude and phase of the low-pass filter's response.

Firstly, the final expression indicates that the sensor has a square-law response to changes in contrast. Secondly, there is a sinusoidal relationship with the orientation of the stimulus. The algorithm gives a large positive response to stimuli in the preferred direction, a large negative response to null direction, and zero response to orthogonal orientations. Thirdly, in response to the sinusoidal grating, the spatial frequency response peaks when  $f_s \cos \theta = 0.25$  cycles/pixel. Lastly, the temporal properties of the sensor can be extracted from the computed output by analyzing the terms that include temporal frequency  $P(f_t)^2 H(f_t) \sin(\phi_t(f_t))$ . According to these terms, the temporal frequency tuning peaks at the cutoff frequency of the low-pass filter and the overall responses exhibits an envelope which corresponds to the temporal frequency tuning of the photoreceptor.

#### 4.3.2.4 Multi-chip implementation of the model:

The multi-chip implementation of the algorithm is composed of the sender and receiver chip pair and illustrated in Figure 4.15. The details of the sender is given in Section 4.2. The receiver chip includes an AER circuitry and 2D array of motion computing pixels.

As shown in the energy model (Figure 4.13), two different temporal filters are introduced to each signal. This stage of implementation is achieved by employing the integrating circuits in the receiver chip. This circuit provides not only integration but also the necessary delay for motion computation.

In the first version of the implementation shown in Figure 4.16, the output currents from the integration stages are subtracted and absolute valued, and the final output is obtained by sum-

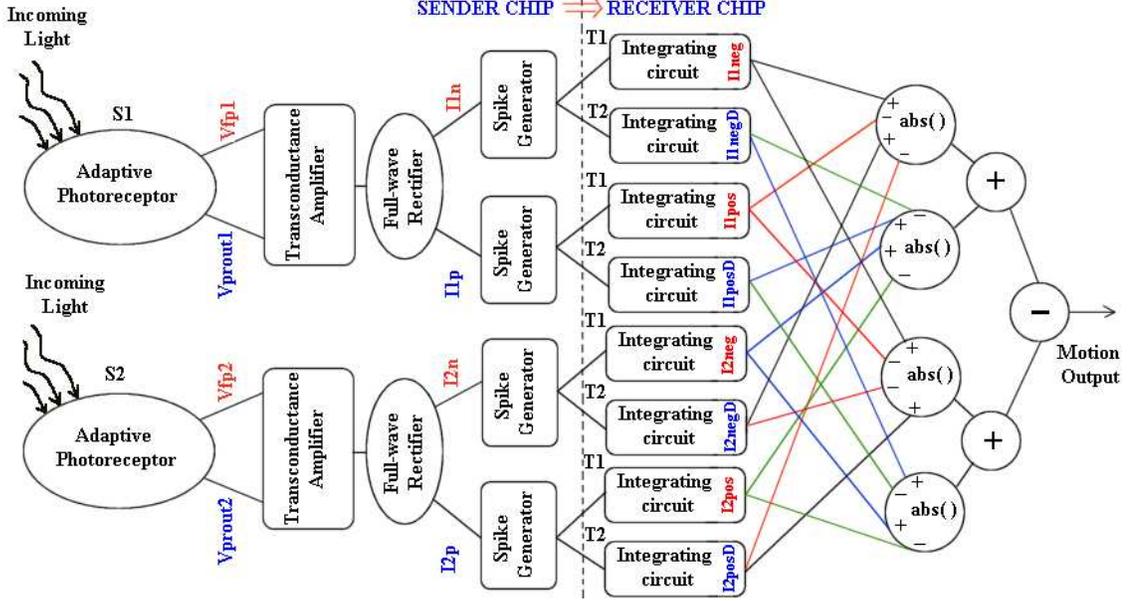


Figure 4.15: The multi-chip implementation of the Adelson-Bergen algorithm.  $V_{prou}$  and  $V_{fb}$  represent the photoreceptor's output and feedback response, respectively. In addition,  $I_p$  and  $I_n$  refer to positive and negative parts of the rectified signal, and in the receiver part,  $I_{pos}$  and  $I_{neg}$  represent the integrated versions of these signals. Lastly,  $I_{posD}$  and  $I_{negD}$  are the signals that are delayed relative to  $I_{pos}$  and  $I_{neg}$ .

ming and subtracting the currents from absolute value circuits. The formulation of the performed computation is shown below.

$$\begin{aligned}
 I_{out} = & |I_{2negD} + I_{1neg} - I_{2posD} - I_{1pos}| + |I_{2neg} + I_{1posD} - I_{2pos} - I_{1negD}| \\
 & - |I_{2posD} + I_{1neg} - I_{2negD} - I_{1pos}| - |I_{2neg} + I_{1negD} - I_{2pos} - I_{1posD}| \quad (4.53)
 \end{aligned}$$

In the second version of the implementation of the model (Figure 4.17), a normalized squaring circuit is added to increase the nonlinearity that can be introduced to the signal. This normalized squaring circuit takes the square of the absolute valued signal. As a result, the final output obtained from the circuit can be shown as follows:

$$\begin{aligned}
 I_{out} = & |I_{2negD} + I_{1neg} - I_{2posD} - I_{1pos}|^2 + |I_{2neg} + I_{1posD} - I_{2pos} - I_{1negD}|^2 \\
 & - |I_{2posD} + I_{1neg} - I_{2negD} - I_{1pos}|^2 - |I_{2neg} + I_{1negD} - I_{2pos} - I_{1posD}|^2 \quad (4.54)
 \end{aligned}$$

The first version of the receiver includes 55 transistors and 4 capacitors in each pixel while the second version of the implementation contains 76 transistors and 4 capacitors. The layouts of the receiver pixel and chip for the first version are shown in Figures A.12 and A.13, respectively. Also, the pixel and chip layouts of the second version are shown in Figure A.14 and Figure A.15. These chips include  $6 \times 6$  pixels.

### 4.3.3 Barlow-Levick model

Barlow and Levick [6] conducted experiments on rabbit retinal ganglion cells by stimulating them with a sequence of discrete illumination steps in two neighboring locations. They found that the

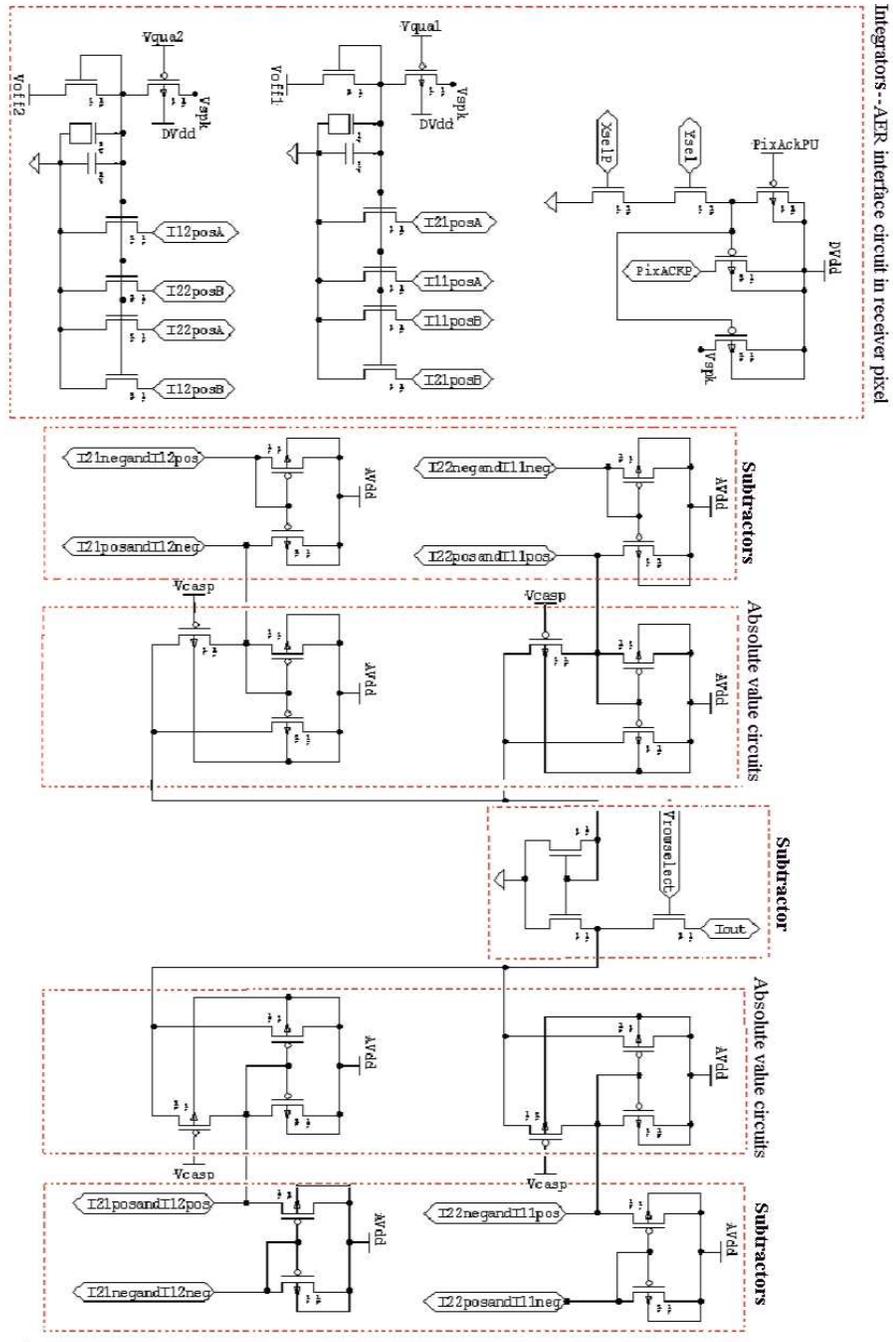


Figure 4.16: The first version of the P pixel in the Adelson-Bergen receiver chip. The main pixel is composed of the P and N pixels. After the integration of the incoming signals in the P and N pixels, the rest of the motion computation is performed in the P pixel. Therefore, the N pixel includes only integrators.

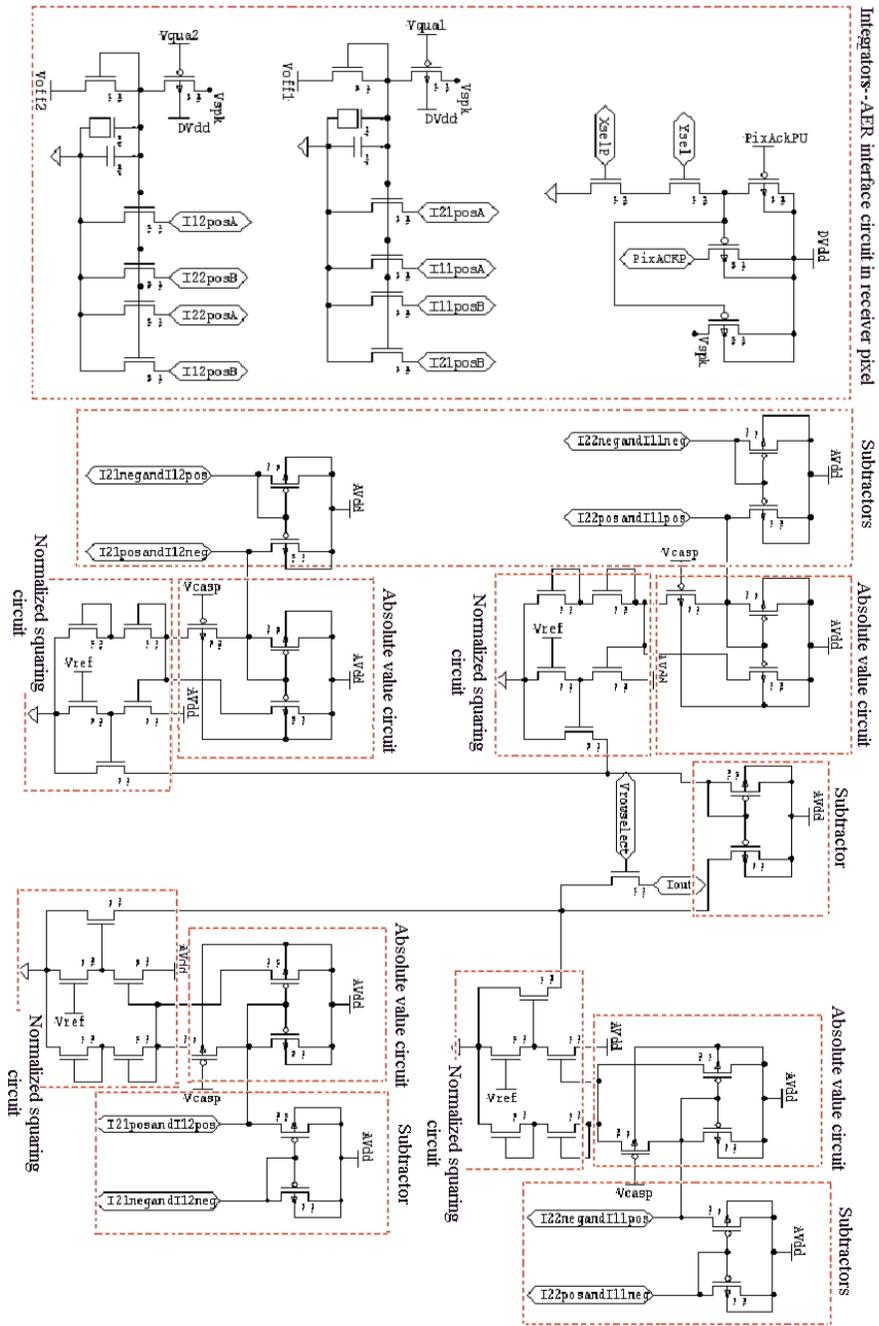


Figure 4.17: The second version of the P-pixel in the Adelson-Bergen receiver chip. As in the first version of Adelson-Bergen receiver chip, the main pixel is composed of the P and N pixels. Both P and N pixels integrate the incoming signals but the rest of the motion computation is performed only in the P pixel.

response to a null direction sequence was significantly suppressed, whereas the response to a preferred direction sequence was estimated to be the sum of the individual responses [10]. By using the findings of the experiments, they explained the direction selectivity in the rabbit retina; it employs inhibitory connections between the direction selective cells to achieve the desired direction selectivity [5].

Benson and Delbruck [7] utilized this null inhibition phenomena to implement a monolithic analog VLSI vision sensor. They used temporal intensity changes to achieve the excitation and inhibition. In this thesis, we present an intensity-based model that achieves the direction selectivity by utilizing a correlation based computation.

#### 4.3.3.1 Canonical form:

The original Barlow-Levick model is illustrated in Figure 3.1. The direction selectivity is attained by the excitatory and inhibitory connections between the photoreceptors and direction selective cells. For instance, when a moving edge passes over the photoreceptors from left to right, the left photoreceptor is excited first, causing its direction selective (DS) cell to fire. After the edge reaches the right photoreceptor, it starts to fire and since it has an inhibitory connection to the left DS cell, the right photoreceptor prevents further output from the left DS cell. In the other case, when an edge is moving in the opposite or null direction (right to left), the activity evoked in the right photoreceptor completely inhibits the left DS cell from firing, thus yields a direction selectivity [7].

#### 4.3.3.2 Modified form:

In order to implement this model in analog VLSI hardware, we modified it adequately as illustrated in Figure 4.18 (R. Deutschmann, unpublished data, 2002). This realization is the same as the modified version of the Reichardt algorithm shown in Figure 4.9 except the multiplication stage is exchanged with the absolute value operation. Multiplication is space consuming in terms of the number of transistors used, whereas the absolute value circuit is well-suited to obtain the nonlinearity needed for motion computation. Therefore we obtain better area usage in this implementation than in the Reichardt implementation.

#### 4.3.3.3 Expected response:

The expected response of the modified form of the Barlow-Levick is similar to the response of the Reichardt algorithm, because the only difference between their modified form is the nonlinearity operation stage. In contrast to the Reichardt model, Barlow-Levick model employs the absolute value circuit to obtain the nonlinearity. As a result, the closed form solution of the modified Barlow-Levick model's response to a sinusoidal stimulus can be found by using Equation 4.36.

#### 4.3.3.4 Multi-chip implementation of the model:

The multi-chip implementation of the model includes a sender and receiver chip pair. The implementation is achieved by utilizing the same sender chip (see Section 4.2) that we used to realize other models. It transfers the small field intensity changes to the receiver chip that contains the motion computation circuitry.

The receiver part includes an AER circuit to communicate with the sender chip and a 2D array of pixels to process the intensity information. After the integration of the incoming spikes by the integrator circuit, the negative and positive parts of the received signals are combined as illustrated in Figure 4.19. This computation can be formulated as follows:

$$I_{out} = |I_{2negD} + I_{1pos} - I_{2posD} - I_{1neg}| - |I_{2pos} + I_{1negD} - I_{2neg} - I_{1posD}| \quad (4.55)$$

This equation is equivalent to the formula shown below:

$$I_{out} = |I_1 - I_{2D}| - |I_2 - I_{1D}| \quad (4.56)$$

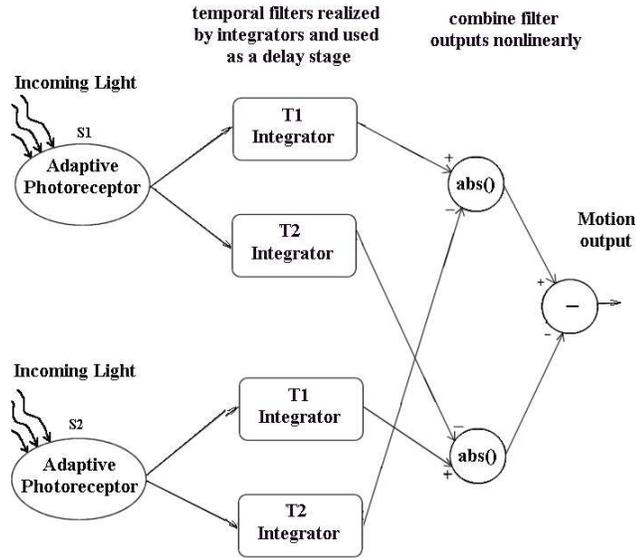


Figure 4.18: Modified version of the Barlow and Levick model.

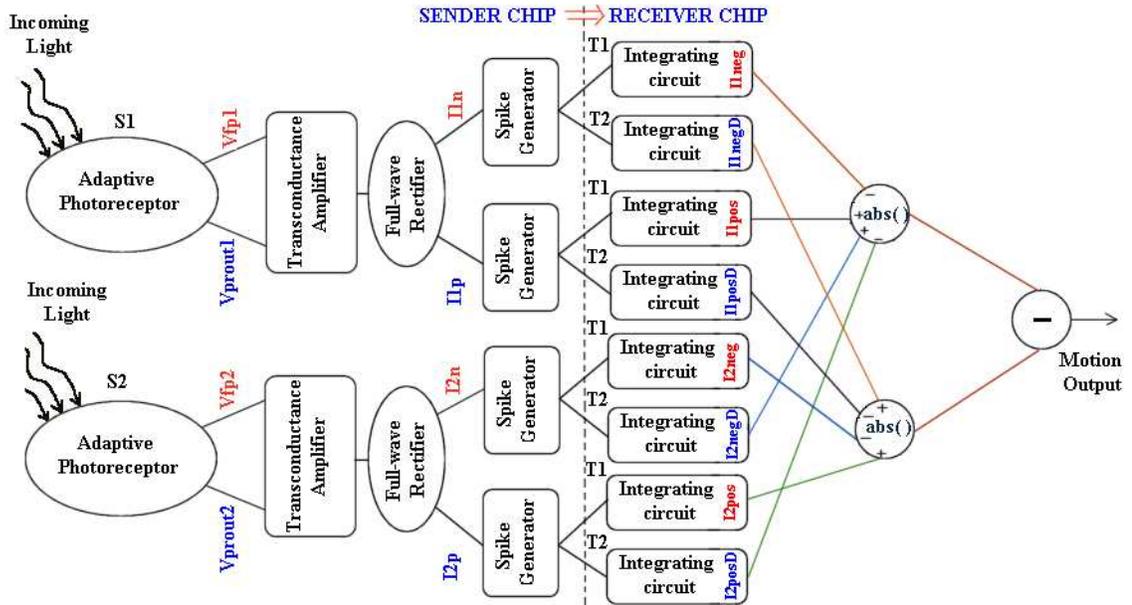


Figure 4.19: Multi-chip implementation of the Barlow and Levick model.

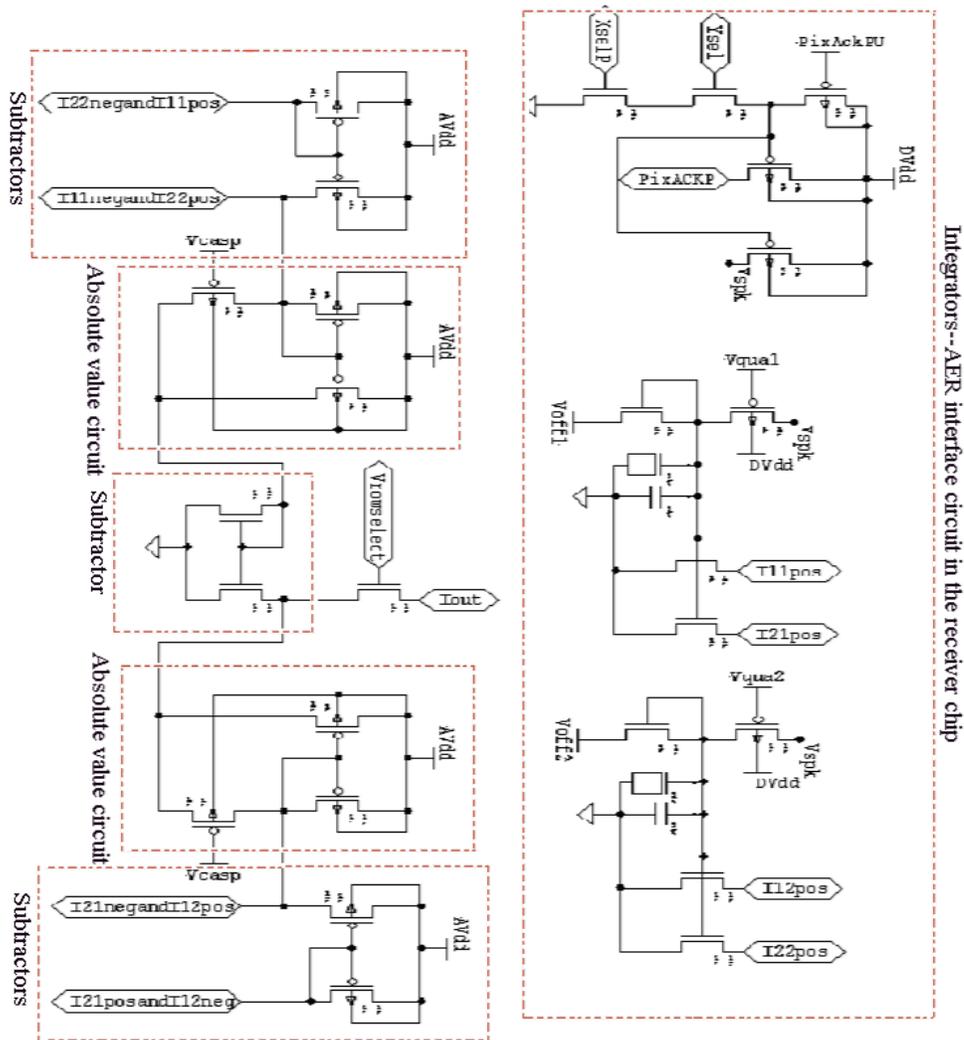


Figure 4.20: The P pixel in the Barlow-Levick receiver chip. The main pixel is composed of the P and N pixels. The N pixel includes only an AER interface circuit and integrators, whereas the P pixel consists of an AER interface circuit, integrators and rest of the motion computation circuitry.

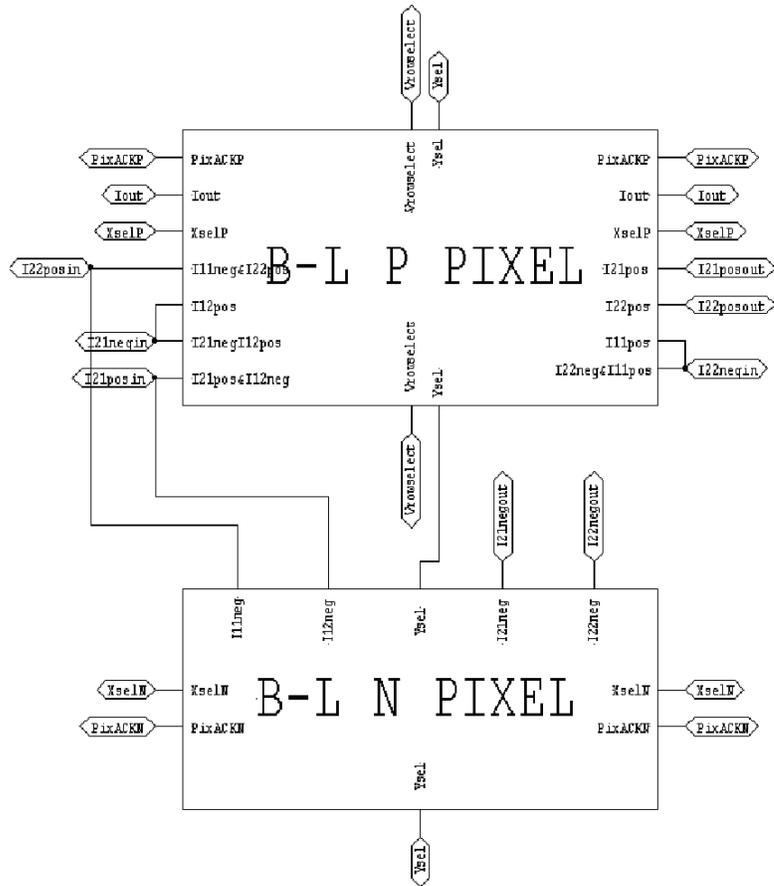


Figure 4.21: The structure of the main pixel in the Barlow-Levick receiver chip.

The schematic of the circuit is shown in Figure 4.20, and as shown, the computation is performed in the P-pixel by combining the integrated spikes in the P and N pixels (Figure 4.21). The pixel level implementation of the Barlow-Levick model contains 34 transistors and 4 capacitors. In comparison to the Reichardt and Adelson-Bergen models, this implementation contains fewer transistors and therefore is more space efficient. The layouts of the Barlow-Levick receiver pixel and chip are shown in Figure A.16 and A.17. The Barlow-Levick chip includes  $6 \times 7$  pixels.

## 4.4 Characterization results

In this section, we present the characterization results of the multi-chip implementations of the algorithms. The experiments were performed by using computer-generated sinusoidal grating stimuli on an LCD screen. In the experiments, while one parameter was varied, the others were held constant. In order to remove the phase dependence of the sensor and prevent artifacts, output voltages were averaged over 10 temporal periods of the stimulus. The output of the sensor was obtained in current mode and converted to voltage by utilizing a current sense amplifier with a 3.9 megohm feedback resistor.

The first experiment was performed by using a sinusoidal grating stimulus to test the direction selectivity of the sensors for preferred, null, orthogonal motion, and no-motion cases. As can be seen in Figure 4.22, the responses of the sensors quite clearly prove the direction selectivity. In case of an orthogonal motion stimulus, all sensors responded with zero average response.

In the second experiment (Figure 4.23), it is observed that all sensors show sinusoidal dependence to an orientation sweep of a sinusoidal grating as expected from the theoretical results of the algorithms. At 90 degrees all the sensors yield a positive peak response and at 270 degrees their outputs reach a negative maximum.

A contrast experiment was achieved by changing the contrast of the sinusoidal stimulus from 0 to 100 percent. This experiment was repeated 100 times for each sensor and in Figure 4.24 the mean outputs of these experiments are illustrated. All the sensors responded strongly down to 10 percent contrast and direction selectivity was possible even lower. The Adelson-Bergen and Barlow-Levick sensors showed better response compared to the Reichardt sensor. The reason might be the multiplication circuit in the Reichardt receiver pixel. The positive and negative pixel circuits have different  $V_{mulbias}$  biases (Figure 4.11). This bias is a normalizing bias employed in the four quadrant multiplier (explained in Section 2.8). Any mismatch between bias transistors or a difference in the value of the bias may cause problems for low-contrast levels. Also, this circuit is assumed to work as a four quadrant multiplier only if the resistances  $R_1$  and  $R_2$  are equal. In the implementation, we used a diode connected transistor to realize  $R_1$  and a normal transistor to realize  $R_2$  in order to subtract the currents from P and N pixels. If there is a mismatch between these transistors then the final output deviates from the expected result. The transistor mismatch is very probable for low current levels which correspond to low contrasts. Hence we can say that the main reason for the Reichardt sensor's unexpected response is the mismatch between the transistors that realize the resistors. This sensor can be improved by using diode connected transistors for both resistances. Alternative to this solution, employing one quadrant multiplier instead of using four quadrant multiplier can be a better solution since minimizing transistor mismatch is a difficult problem. The circuit shown in Figure 2.9a is a good candidate for this purpose.

In the last two experiments, the sensors were tested to acquire the spatial and temporal frequency characteristics of the implementations. The responses of the sensors to a temporal frequency sweep is shown in Figure 4.25. The output of the Adelson-Bergen sensor peaks at around 1 Hz, the Reichardt sensor peaks around 2Hz and the Barlow-Levick sensor reaches to its peak response around 3Hz. The tuned temporal frequencies of the sensors can be changed by altering the voltage values of the integrating receiver circuit bias  $V_{qua}$  as illustrated in Figure 4.26. Besides, it is obvious from the temporal frequency response that the sensors can detect the motion for a velocity range of more than one order of magnitude. These responses justify the use of the integrating circuit as a temporal filter in the motion computation. Moreover, the responses of the sensors to a spatial frequency sweep are illustrated in Figure 4.27. The plots of the multi-chip sensors peak around

0.4 cycles/pixel and the sensors show a strong spatial aliasing around 0.9 cycles/pixel. Lastly, in Figure 4.28, the spatiotemporal responses of the sensors are illustrated. These plots show the mean output of the sensors when they are simulated by sinusoidal gratings varying in both spatial and temporal frequency. The mean output is plotted for spatial frequencies on the X-axis versus temporal frequencies on the Y-axis. It is clear from the graph that the models respond best to a particular spatiotemporal frequency for which they are tuned and their response decreases at other frequencies.

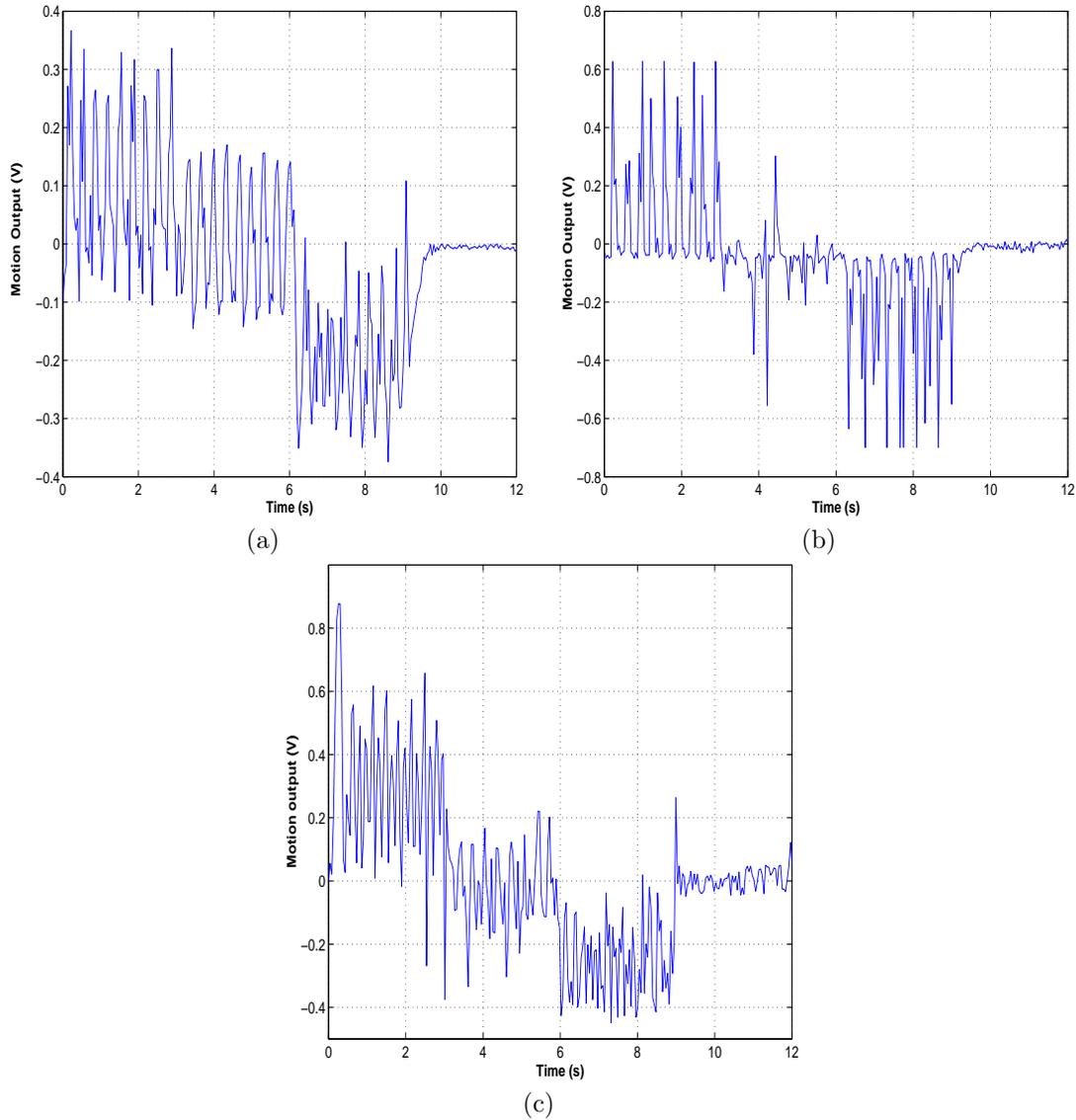


Figure 4.22: Raw temporal output of the motion sensors. These outputs are not averaged and represents the real time single pixel response of the sensors. In the interval of 0-3 seconds a sinusoidal stimulus is presented in the preferred direction. Between 3-6 seconds a sinusoidal stimulus moving orthogonal to the sensor orientation is presented. After that, the sensor is exposed to a null direction sinusoidal stimulus. Lastly, no stimulus is presented between 9-12 seconds. (a) Raw data of the Adelson-Bergen sensor.(b) Raw data obtained from the Reichardt sensor. (c) The Barlow-Levick sensor's raw data.

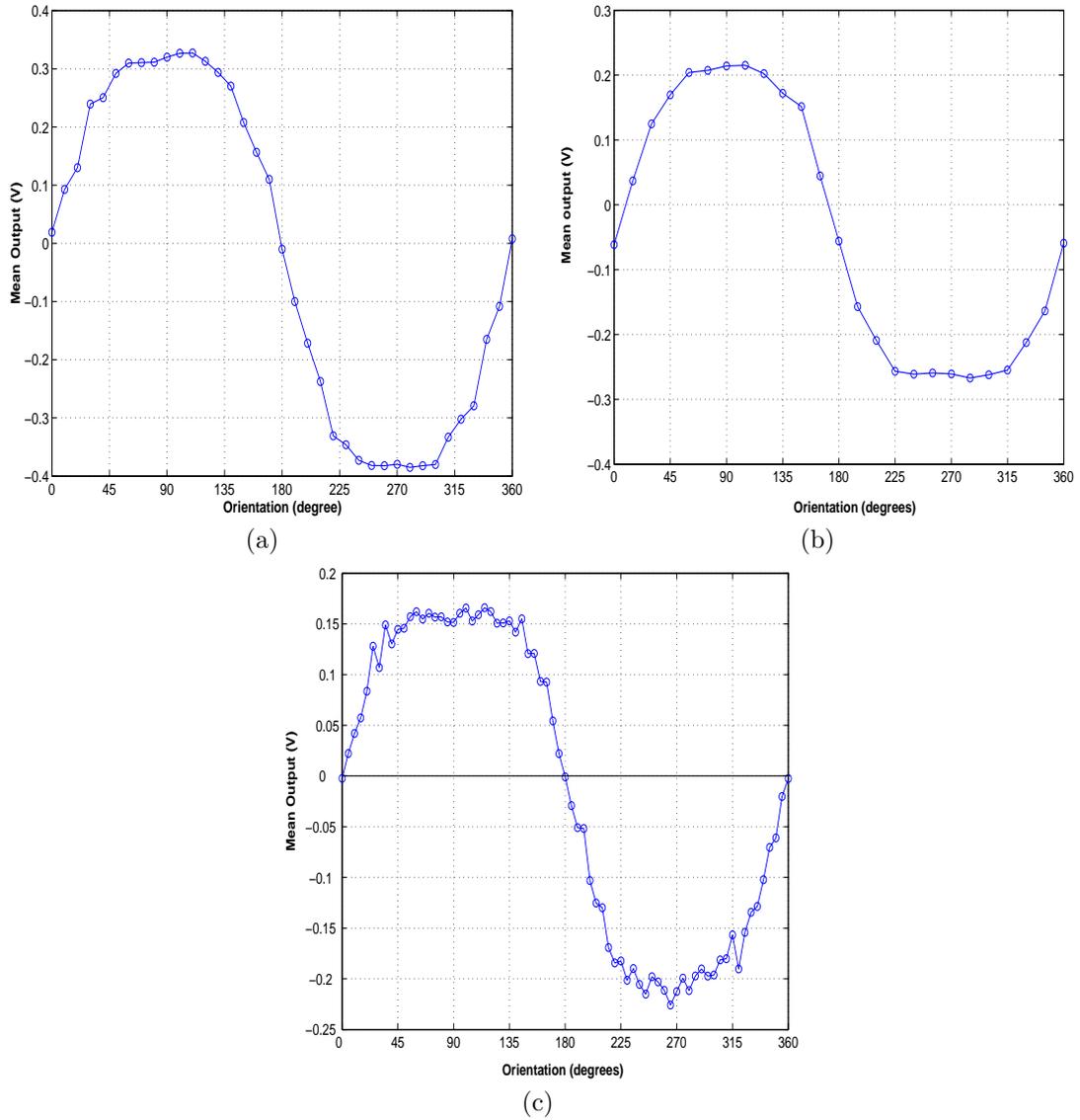


Figure 4.23: Orientation sweep of a sinusoidal stimulus. The responses were averaged and therefore represent the mean responses of the sensors. The stimulus was presented at varying directions relative to the motion sensor, which is optimally oriented for a stimulus at 90 degrees. (a) Orientation sweep response of the Adelson-Bergen model. (b) Response of the Reichardt sensor to orientation sweep. (c) Orientation sweep response of the Barlow-Levick model.

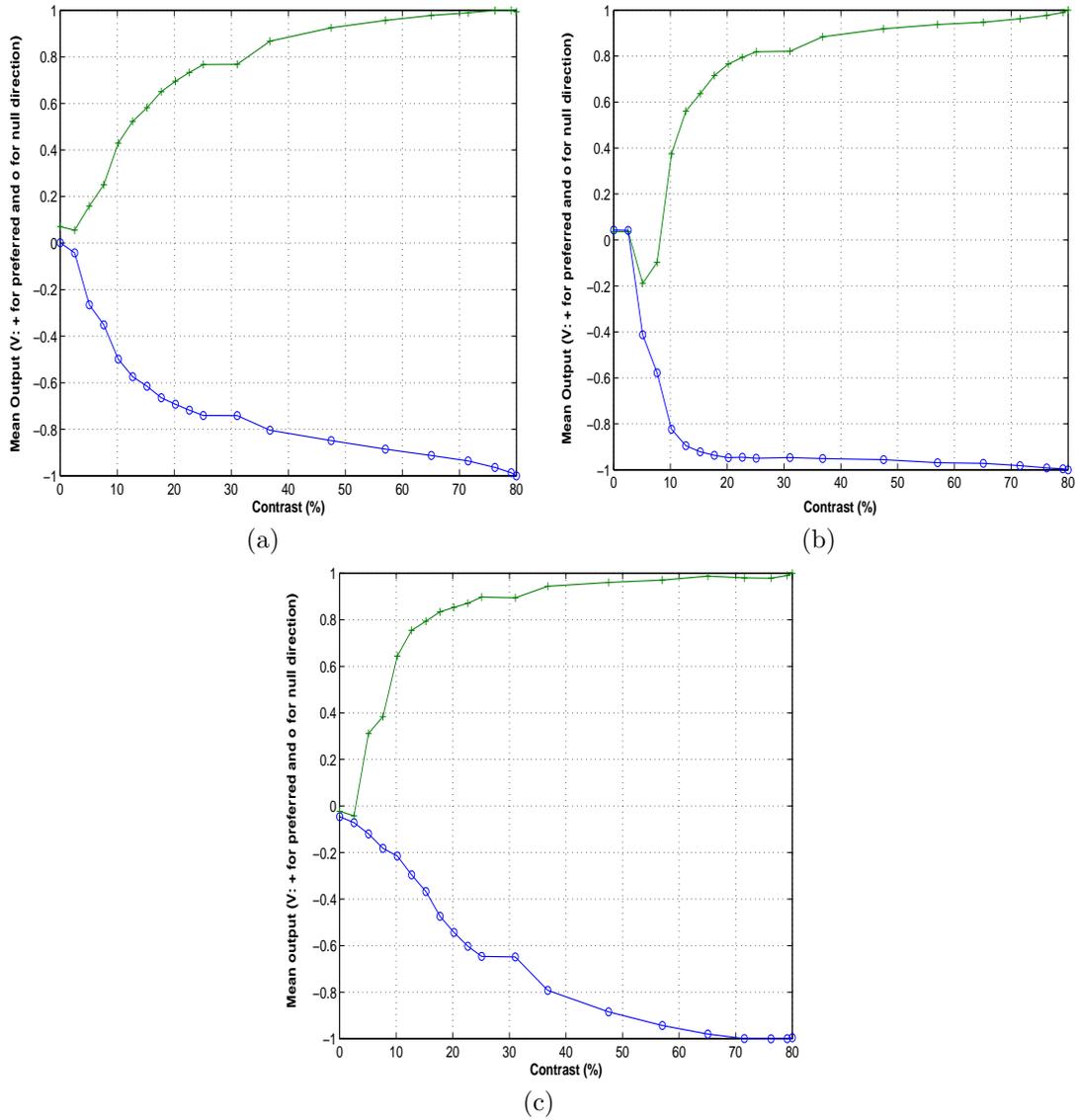


Figure 4.24: Contrast sweep of a sinusoidal stimulus. The responses represent the mean output of the sensors. (a) The response of the Adelson-Bergen sensor. (b) The response of the Reichardt sensor. (c) The response of the Barlow-Levick sensor.

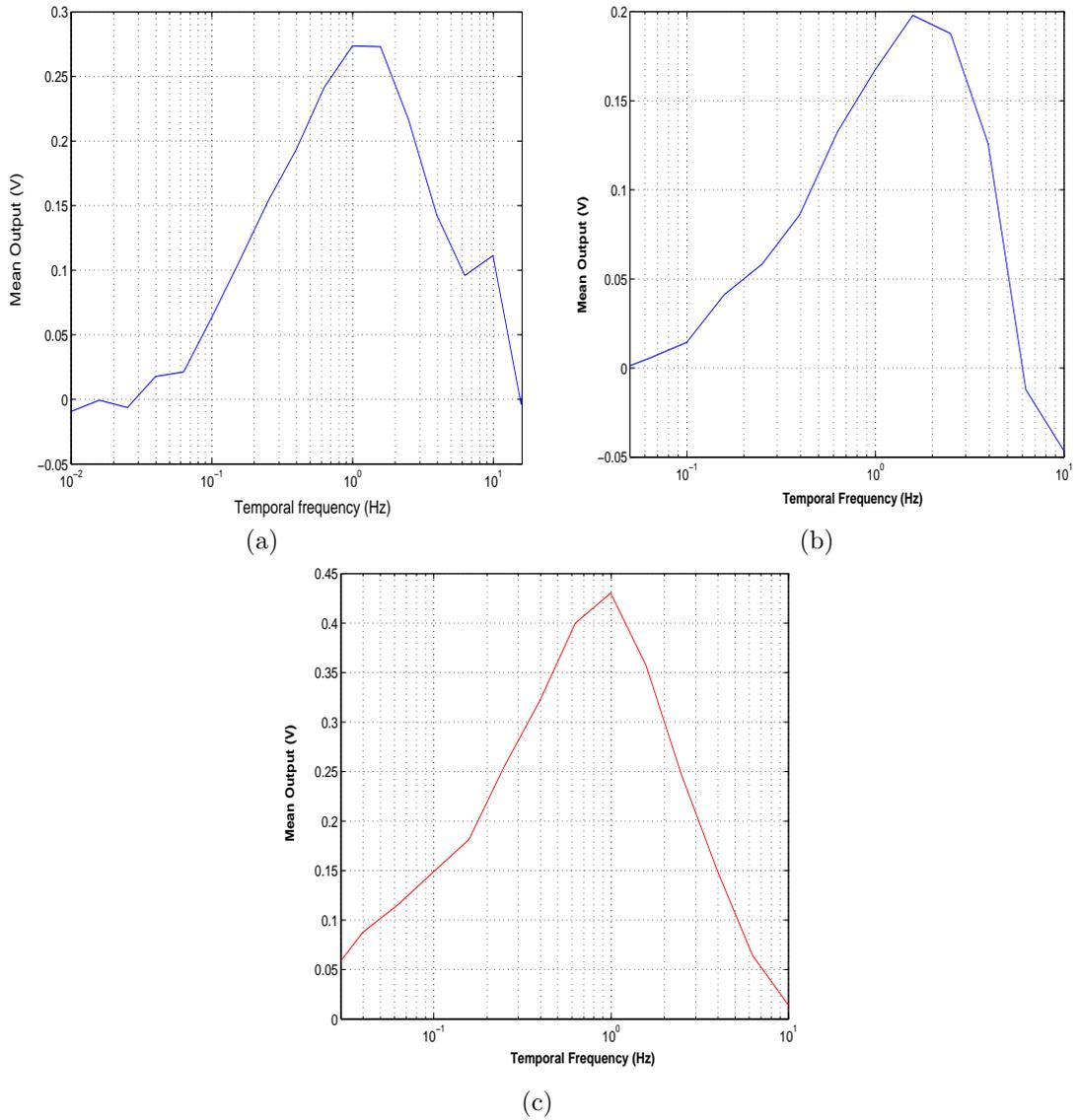


Figure 4.25: Temporal frequency sweep of a sinusoidal stimulus. The responses represent the mean output of the sensors. (a) The response of the Adelson-Bergen sensor. (b) The Reichardt sensor response. (c) The response of Barlow-Levick sensor.

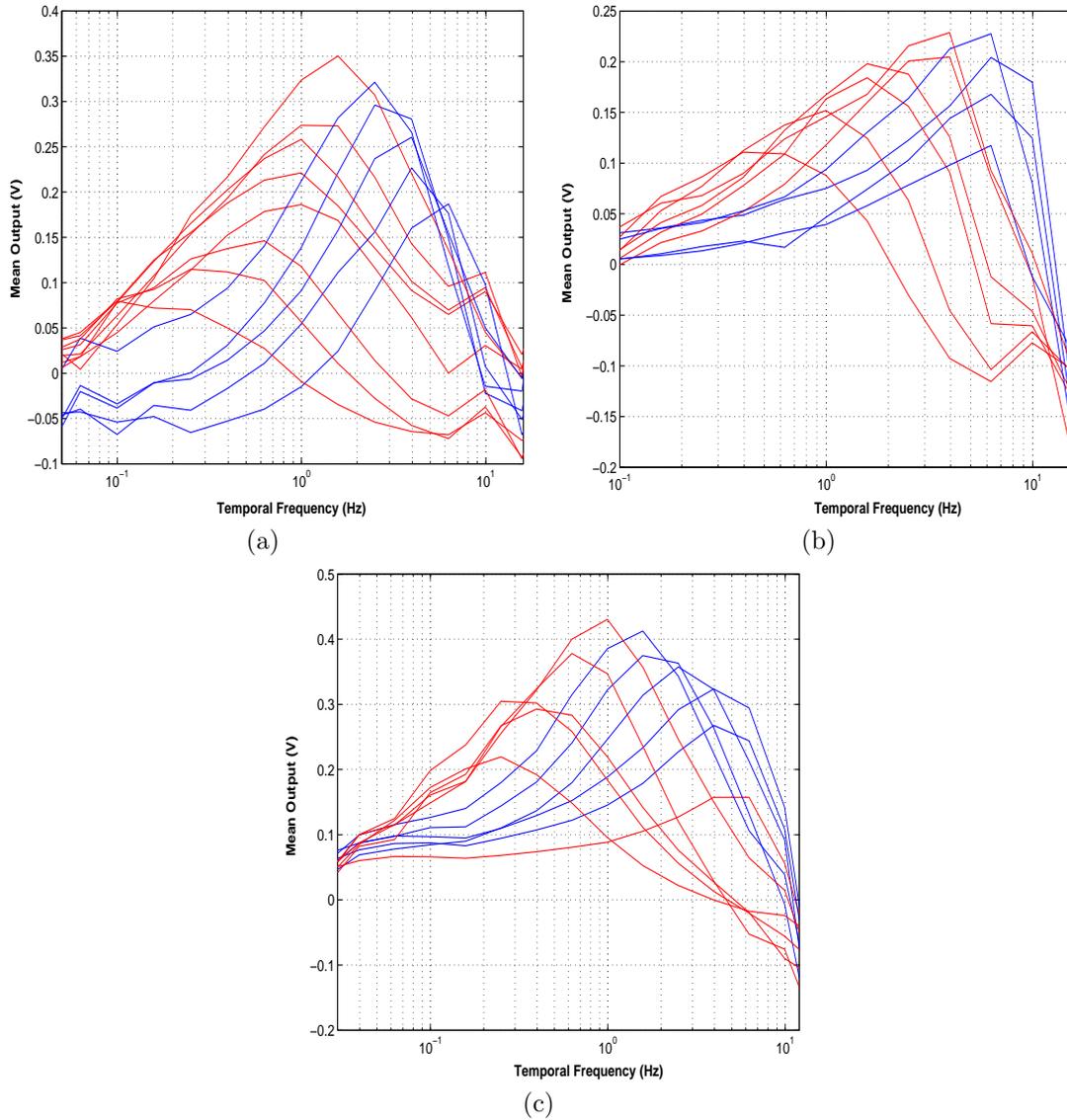


Figure 4.26: Variation of temporal frequency tuning. By changing the bias  $V_{qua}$ , it is possible to change the temporal frequency tuning of the sensors. Each curve in these plots are at a different bias setting. The overall envelope of all plots is the photoreceptor temporal frequency response, the low frequency cutoff of which is due to adaptation, and the high frequency cutoff of which is due to the capacitance driven at its output and its biases.

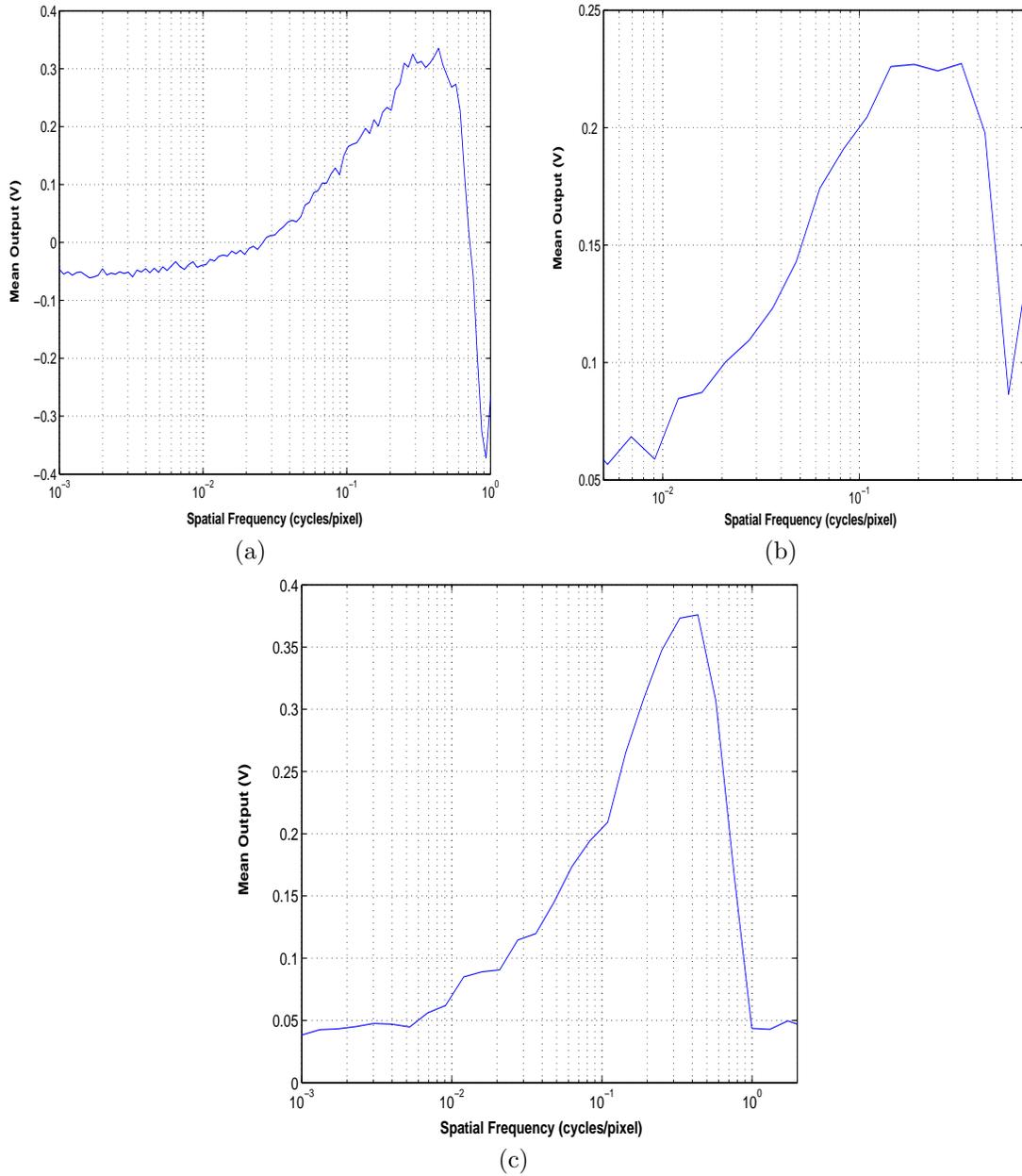


Figure 4.27: Spatial frequency sweep of a sinusoidal stimulus. The responses represent the mean output of the sensors. (a) The response of the Adelson-Bergen sensor. (b) The Reichardt sensor response. (c) The response of Barlow-Levick sensor.

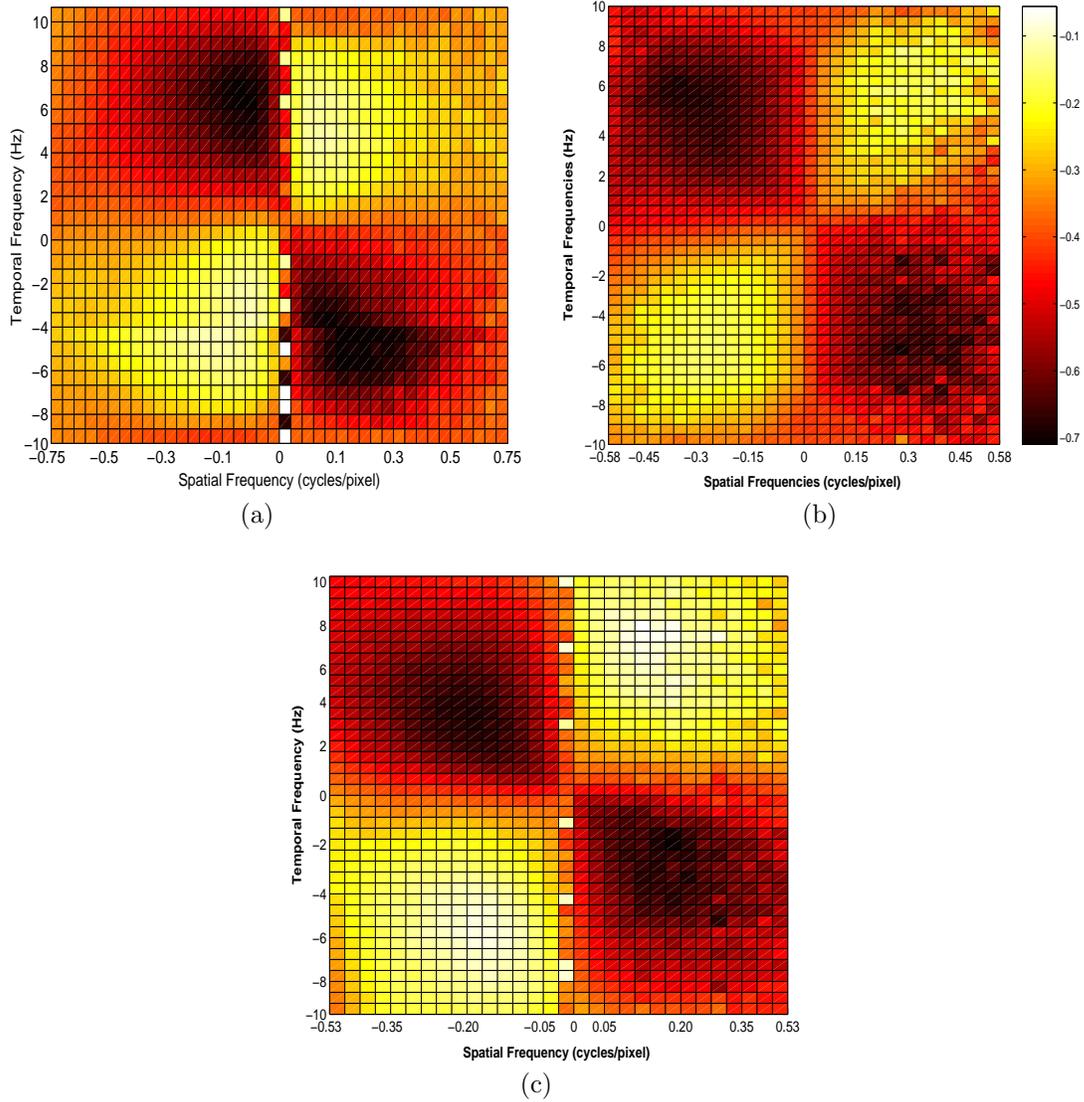


Figure 4.28: Spatiotemporal frequency plots of the sensors. These spatiotemporal responses represent the mean output of the sensors. The mean for specific spatial and temporal frequencies is obtained by averaging the sensor responses in time. Light colors indicate positive and dark colors indicate negative average response. (a) The response of the Adelson-Bergen sensor. (b) The Reichardt sensor response. (c) The response of Barlow-Levick sensor.

## Chapter 5

# Obstacle Avoidance Algorithm

A variety of techniques and implementations have been proposed to solve the problems faced in navigation in an environment cluttered with obstacles. In the implementation of a biologically inspired robot that can navigate in complex environments, Franceschini et al. [27] utilized the fact that during pure translations, the optical flow generated by the ego-motion of the observer and the movements of other objects contains information about the distance of these objects relative to the observer. The problem with this implementation is that when the visual axis is close to the heading direction, the system becomes inefficient in detecting the objects right in front of it; this problem is known as the parallax blind spot problem. Later, Mura and Franceschini [59] presented a scanning movement behavior as a solution to this problem.

As an alternative to scanning movement behavior, zigzag behavior was suggested to be an effective way of dealing with the parallax blind spot problem. This behavior was inspired by observations obtained from experiments on some insects and lower vertebrates [69], [82]. Firstly, Sobey [70] used this behavior in his obstacle avoidance implementation. In his system, the optical flow is collected for 16 frames for each discrete movement and the decision for the next movement is made on the basis of the distance of an identified object. Secondly, Lewis [53] utilized this strategy in his biologically inspired system by integrating it with information gleaned from both the behavior of bees and the anatomy of flies. This system exploits the findings of experiments by Srinivasan and Zhang [73] on the movement detection system of bees. These experiments elucidated the fact that the centering response of bees is mediated by a direction-insensitive movement system. Accordingly Lewis [53] employed the principles of the non-direction-selective motion sensor into his implementation to build a robust biologically inspired system that can navigate between obstacles.

In another implementation by Huber and Bulthoff [39], an artificial agent was developed by integrating biological principles with the capabilities of genetic algorithms. In this system, the computation is performed by using visual input from 360 degrees of the visual field. After computing the optical flow field, the motion outputs of the sensors are spatially weighted by a gain function that is optimized by a genetic algorithm. As a result of the experiments, they concluded that the genetic algorithm yields its maximum sensitivity for a frontolateral visual field. By integrating this structure with the genetic algorithm, they succeeded in obtaining a generation that could navigate in complex environments.

The lobula giant movement detector (LGMD) and its postsynaptic partner the descending contralateral movement detector (DCMD) are the neurons located in the visual system of the locust. These neurons are wide-field visual interneurons that can trigger escape response to objects approaching on a collision course. Blanchard et al. [8] conducted experiments with a mobile robot to investigate collision avoidance behavior by using a model based on the neuronal responses of the LGMD neuron in locusts. The escape behavior is initiated when the angle subtended by a looming object exceeds a certain threshold. Also, Indiveri [40] presented an analog VLSI architecture that implements a functional model of the DCMD neuron.

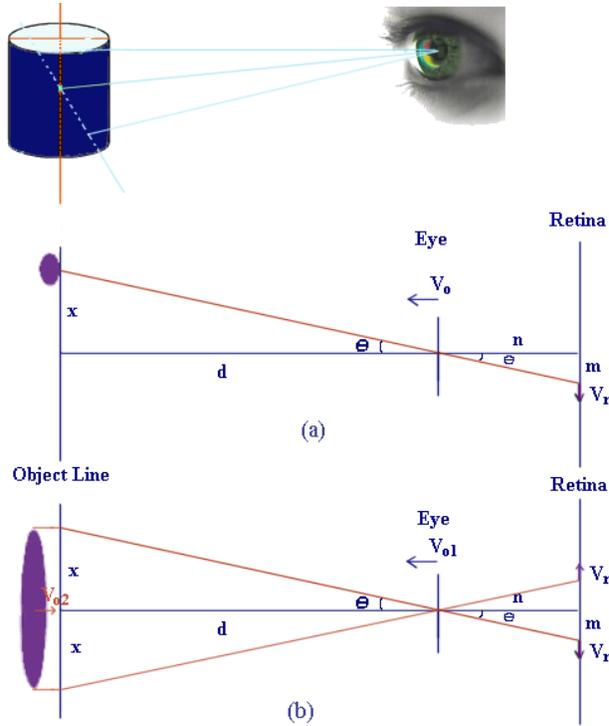


Figure 5.1: Motion parallax. (a) The effect on the retina of egomotion towards an object is illustrated. (b) The effect of egomotion and/or object motion on a collision course is depicted.

## 5.1 Algorithm

In this section we describe the main components of the algorithm that we utilized to build the obstacle avoidance system on a simulated agent. Firstly, we used motion parallax to obtain the properties of the optical motion field created by the agent's own motion and/or the motion of objects in the visual field. Secondly, we made use of the information obtained from physiological and behavioral experiments conducted on bees to construct the centering behavior. Lastly, we employed findings obtained from locust and pigeon anatomy and physiology in order to build a collision sensitive system.

### 5.1.1 Motion Parallax

In the case of pure translation, motion parallax provides reliable information on the structure of the environment since retinal image motion yields larger responses for closer objects relative to distant objects. This information has been utilized in different implementations to build artificial systems that can navigate successfully. For instance, Franceschini et al. [27] employed motion parallax computation in their artificial obstacle avoidance system by evaluating the optical flow on a horizontal 360 degree ring sensor. In this study we use this fact to compute the relative distance of objects in order to ensure a safe navigation path for the simulated agent. We also utilize it in a time-to-contact computation to avoid collisions with objects in the heading direction of motion.

In Figure 5.1a, the effect of an object's motion on the retina while the observer moves is illustrated. In this case, the object is not in the heading direction. Hence in order to avoid collision with such an object we have to relate the relative distance and the speed of the observer to those of the object. In this figure  $\theta$  is defined as the angle subtended by the object from the line of motion,  $x$  represents the distance of the object perpendicular to the line of the heading direction and  $d$  is used to show the distance of the observer perpendicular to the trajectory of the object. Also  $V_o$

represents the speed of the observer and  $V_r$  is used to show the speed of the object's image on the retina. In addition,  $m$  represents the distance of the object's image on the retina perpendicular to the line of the heading direction and  $n$  is used to show the length of the retinal focus.

In Figure 5.1a, the relation between the distance parameters and the angle can be found as follows:

$$\theta = \tan^{-1} \frac{x}{d} = \tan^{-1} \frac{m}{n} \quad (5.1)$$

The time derivative of  $\theta$  gives some idea of the relative change in the distance parameters. In this study, we assume that the movement of the observer is a pure translation in the heading direction at a constant speed and therefore  $V_o$  can be set equal to the time derivative of the distance  $d$  and  $V_r$  can be set equal to the time derivative of the distance  $m$ .

$$\frac{\partial \theta}{\partial t} = \frac{\frac{-x}{d^2}}{1 + (\frac{x}{d})^2} \frac{\partial d}{\partial t} = -\frac{1}{d} \tan \theta \frac{1}{1 + \tan^2 \theta} V_o \quad (5.2)$$

$$\frac{\partial \theta}{\partial t} = \frac{\frac{1}{n}}{1 + (\frac{m}{n})^2} \frac{\partial m}{\partial t} = \frac{1}{n} \frac{1}{1 + \tan^2 \theta} V_r \quad (5.3)$$

where  $x$  and  $n$  are assumed to be constant and,  $V_o = \frac{\partial d}{\partial t}$  and  $V_r = \frac{\partial m}{\partial t}$ . By equating these two functions and ignoring the minus sign in equation 5.2, we can find the following expression:

$$V_r = \frac{m}{d} V_o = \frac{nx}{d^2} V_o \quad (5.4)$$

These equations provide certain relationships between the distance and the speed parameters. From these relations, we can easily derive the distances  $x$ ,  $d$  and  $z$  (the distance of the object from the observer) as follows:

$$x = \frac{m^2 V_o}{n V_r} \quad (5.5)$$

$$d = m \frac{V_o}{V_r} \quad (5.6)$$

$$z = \sqrt{x^2 + d^2} = \frac{V_o m}{V_r n} \sqrt{m^2 + n^2} \quad (5.7)$$

From these equations, the relative distance parameters can easily be calculated without knowing the external distance parameters such as  $d$  and  $x$ . In addition,  $m$ , the distance of the retinal image motion from the centerline can be easily incorporated into any calculation since it can be extracted from the sensed retinal motion.

The optical flow computation can be performed with an array of motion sensors, and with enough resolution, accurate behavioral responses can be produced. The results attained from this analysis are useful in particular for centering behavior to avoid collisions by selecting a path equidistant from nearby objects.

In the case when an agent is supposed to avoid collisions with obstacles or initiate an escape response, Figure 5.1b can be utilized to illustrate the scenario and to perform mathematical analysis. As in the previous case,  $\theta$  is defined as it is formulated in Equation 5.1. From this formula, the relation between the distance parameters and the velocity information can be acquired as in the first case. In contrast to the first case, in the second case  $x$  represents the radius of the object. In addition to this, in the previous case the main parameter was  $z$ , the distance of the object from the observer, but now the crucial parameter becomes  $d$ , the distance of the observer from the center of

the object.

By using the relations between the speed and the distance parameters, we can calculate the distance of the observer from the center of the object as follows:

$$d = m \frac{V_o}{V_r} \quad (5.8)$$

The time-to-collision with the object can be found as follows:

$$t_{collision} = \frac{d}{V_o} = \frac{m}{V_r} \quad (5.9)$$

In this equation,  $m$  and  $V_r$  can be extracted from the optical motion fields on the retina. By computing these parameters, the time-to-collision can be easily obtained.

### 5.1.2 The Centering Behavior

The work done with bees by Srinivasan and Zhang [74] elicited the fact that in their visual course control bees balance the speeds of image motion on their two eyes (such as when navigating through a narrow gap). The most crucial point in this kind of behavior is that bees maintain an equal distance from objects in their environment by balancing the apparent angular speeds on their two eyes. In addition, they balance the image speeds independent of the contrast or spatial frequency of the objects on the two sides.

The underlying mechanism for the centering behavior is a non-direction selective movement detector, output of which encodes retinal velocity information [73], [74]. Unlike the direction selective system that is used in motion computation for optomotor responses, the direction insensitive pathway has functions in obstacle avoidance and centering behavior. Moreover, it has been found that the non-direction selective movement system has a higher frequency response relative to the direction selective pathway [72].

It is suggested that the centering mechanism may also play a role in the movement avoidance response [71], [75]. This kind of behavior is observed in bees, and they use it to avoid flying toward moving objects. Thus the centering response can be used to explain movement avoidance as well as obstacle avoidance.

In implementations on robots or different agents, the main aim is to maximize the robustness and effectiveness of the obstacle avoidance system. The centering behavior provides a way of accomplishing this feat. It can be successfully incorporated into systems to minimize the possibility of collisions. Although Srinivasan et al. [74] state that bees balance the angular speed of images perceived by their two eyes, this statement is not sufficient to determine the neural computation involved in this balancing procedure. Furthermore, retinal image motion can be processed in different ways to balance the apparent angular speed on the two eyes. For example, in a cluttered environment, bees navigate between objects which are at different distances from them and which therefore cause different motion responses on their retinas. Even if the motion vectors on both eyes are different by two orders of magnitude, centering behavior is still achieved by bees.

In this study, the centering response is achieved with pure translations. The main reason for this choice is that in order to realize a robust collision avoidance system, a purely translatory motion signal is needed [20], [27], [85]. The motion contrast information is extracted from the right and the left optical motion fields on the two eyes. This information is used as a gain in the computation of the next rotatory movement of the system. In this way, the effect of a larger motion on one of the eyes can be suppressed with a motion in opposite direction. Assuming that  $I_{right}$  and  $I_{left}$  are the angular speed of image perceived by the right and left eye respectively, then motion contrast can be expressed as follows:

$$I_{contrast} = \frac{I_{right} - I_{left}}{I_{right} + I_{left}} \quad (5.10)$$

$I_{contrast}$  not only provides information on the relative magnitude of  $I_{right}$  and  $I_{left}$  but also gives an indication whether the motion is in the preferred or null direction. Its response can be saturated for a range of values and the linearity range can be adjusted accordingly if it is used with a nonlinear function as shown below:

$$I_{gain} = I_{bias} \tanh(C_{linearity} \cdot I_{contrast}) \quad (5.11)$$

where  $C_{linearity}$  is a coefficient used to adjust the linear region of the  $\tanh$  function. The contrast and gain functions are illustrated in Figure 5.2. We utilize the contrast of the right and left motions to determine the amount of the rotatory movement. If the leftward motion is bigger than the rightward one, the gain will be used to determine the amount of the rotation to the right, and vice versa. By using these equations, different design choices can be investigated to build a robust artificial system. In this thesis, we studied three different design strategy to be able to determine the best way of representing the angular speeds on both eyes. For this purpose, we employed a 1D array of elementary motion detectors (EMDs) that are based on Adelson-Bergen algorithm (Section 4.3.2) to obtain the small field visual motion response on both eye. The motion response of this algorithm depends on the spatiotemporal frequency of stimulus. Therefore, the Adelson-Bergen visual motion sensor yields its maximum motion output when the spatiotemporal frequency of stimulus matches the spatiotemporal tuning of the sensor.

Firstly, an obstacle avoidance system can be developed by utilizing the fact that the maximum velocity signals the closest object to each eye. In this way the system is made to focus on imminent collisions. In this thesis, since we use Adelson-Bergen sensor instead of a velocity sensor we utilize the spatiotemporal response of the sensor to employ this obstacle avoidance method. By using EMD responses and the maximum operation, we can express  $I_{right}$  and  $I_{left}$  in Equation 5.10 as follows:

$$I_{right} = \max(I_{EMDr1}, I_{EMDr2}, \dots, I_{EMDrn}) \quad (5.12)$$

$$I_{left} = \max(I_{EMDl1}, I_{EMDl2}, \dots, I_{EMDln}) \quad (5.13)$$

where  $I_{EMDr_i}$  and  $I_{EMDl_i}$  represent the motion outputs of the EMDs on the right and left eyes, respectively, and  $n$  is the number of EMDs on each eye. Using the maximum velocity in navigation may cause sudden turns or unexpected responses. Also, even if an object is in front of the agent, since the motion detection in the heading direction is not too effective in signaling alone motion component of that object, the response will not be reliable. However, it is still very effective particularly in dealing with navigation in cluttered environments, because it focuses only to the closest objects. In this way, the centering task can be reduced to the navigation through the closest objects.

Secondly, the spatial average of angular speeds on both eyes can be employed to compute the motion contrast and gain. In order to obtain the average of the motion outputs on both eyes, we use a predetermined threshold. In this way, we can determine the number of EMDs that have motion responses exceeding this threshold. At the same time, we sum the motion responses that exceed the threshold. By using these parameters we can determine  $I_{right}$  and  $I_{left}$  as follows:

$$I_{right} = \frac{(\dots I_{EMDr_{i-1}} + I_{EMDr_i} + I_{EMDr_{i+1}} + \dots)}{k} \quad (5.14)$$

$$I_{left} = \frac{(\dots I_{EMDl_{j-1}} + I_{EMDl_j} + I_{EMDl_{j+1}} + \dots)}{m} \quad (5.15)$$

where  $I_{EMDr_i}$  is the  $i^{th}$  right EMD response that exceeds the threshold and  $k$  represents the number of responses that exceed the threshold. Similarly,  $I_{EMDl_j}$  is the  $j^{th}$  left EMD response that is larger than the threshold and  $m$  represents the number of responses that have values larger than the threshold. The information obtained from this computation can provide reliable data on the nature of the environment. The averaging operation ensures a safe navigation even in the presence of objects

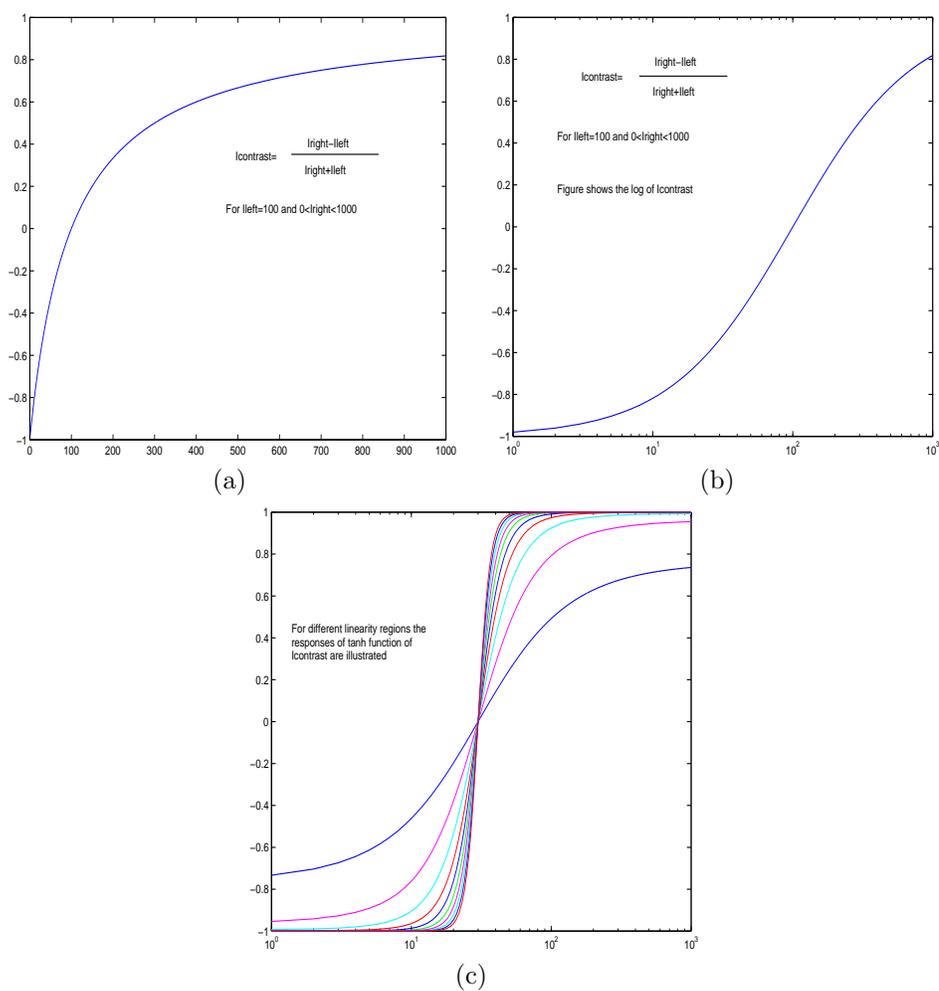


Figure 5.2: Motion contrast. (a)  $I_{contrast}$  is illustrated for the constant values obtained by keeping  $I_{left}=100$  and changing  $I_{right}$  from 0 to 1000. (b) The response of motion contrast is illustrated on a log scale. (c) The response of the  $I_{gain}$  function for different values of  $C_{linearity}$ .

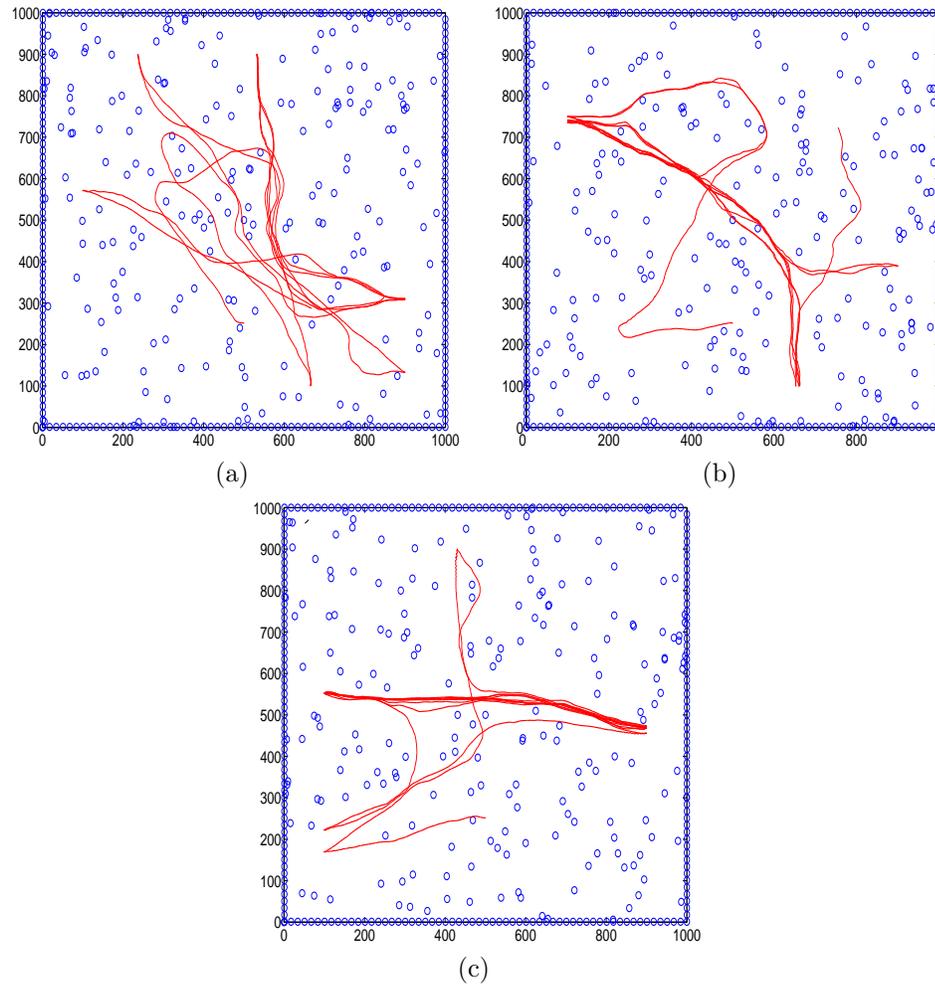


Figure 5.3: Navigation using different centering techniques. (a) Using the maximum motion output in the visual field. (b) Using the average motion output of the optical visual fields on each eye. (c) Using sums of the motion responses on each eye.

that causes different spatiotemporal responses. Although in this kind of computation there has to be a threshold operation, which is not preferable by its nature of operation, this computation can still be employed to realize the centering behavior.

Lastly, the sum of the right and left motions can be used as a reference for the rotatory movements. In this case, the agent tries to avoid the region that causes larger spatiotemporal motion output. In this way, it minimizes the probability of collision since a larger motion response may also mean more objects or closer objects. By using the sum operation, we can express  $I_{right}$  and  $I_{left}$  as follows:

$$I_{right} = I_{EMDr1} + I_{EMDr2} + \dots + I_{EMDrn} \quad (5.16)$$

$$I_{left} = I_{EMDI1} + I_{EMDI2} + \dots + I_{EMDIn} \quad (5.17)$$

In order to obtain quantitative results to compare the effectiveness of these algorithms, we simulated them in 2-D environments. In these simulations, objects are represented as cylinders and the observer is represented as a robot that has 90 degrees of visual field. The size and contrast of the objects are set by their distance from the robot. The arena that the robot navigates in is surrounded by walls and the objects are randomly distributed in it. If the robot navigates out of a predetermined ratio of the arena, a 180 degrees turn is initiated to prevent the robot from leaving the arena. At each step, the robot process the motion created by its own visual motion by using the Adelson-Bergen algorithm and decides upon the next step by applying the algorithms explained above.

After 100,000 steps of simulations for each algorithm we obtained the results shown in Figure 5.3. These figures reinforce the idea that the sum of non-direction selective velocity fields yields the best result. In the first and second figures (Figure 5.3a, Figure 5.3b), it is seen that as the agent navigates, it has 3 collisions in each case, whereas in the last figure (Figure 5.3c), the agent navigates by summing the velocity vectors and has no collision. From these simulations, we can safely conclude that taking the sum of velocity vectors on both eyes is the best way of implementing the centering behavior.

### 5.1.3 Looming Sensitive Neurons and Time-To-Collision Computation

Organisms are anatomically specialized in their environment to cope with complex behavioral tasks. While the largest motion detectors in flies are those responding best to adjust their flight path, the largest motion detector of locusts is dedicated to avoid rapidly approaching objects [27]. Wang and Frost [83] found that in the pigeon brain, the nucleus rotundus responds maximally to objects on a collision course. In this respect, the anatomy of locusts and pigeons provides great deal of information to model neurons specialized in object avoidance and escape mechanisms.

The locust obtains its sensitivity to approaching objects by the neural computations performed by wide-field visual neurons, namely the lobula giant movement detector (LGMD) and the descending contralateral motion detector (DCMD). The DCMD neuron fires when there is a movement anywhere within the visual field, but responds strongly to approaching objects on an impending collision course. Rind et al. [64] proposed that the DCMD tracks edge motion throughout the object approach and produces strongly maintained response when the object moves close to the eye.

The input organization of the LGMD neuron is modelled by Rind and Bramwell [63]. In their work, they claim that the input elements of the LGMD neuron generate excitations and lateral inhibitions when the eye is exposed to the approaching object. Then a critical race between excitation and inhibition determines the response of the neuron. In contrast to this model, Hatsopoulos et al. [32] proposed an equation that assumes a symmetric expansion of the object and produces a response that can be used to avoid collisions. They suggested that the firing rates of the LGMD and the DCMD depend on the product  $\psi(t - \delta)e^{-\alpha\theta(t-\delta)}$ , where  $\theta(t)$  is the angular size subtended by the object during approach,  $\psi(t)$  is the angular edge velocity of the object, and  $\alpha$  is a parameter determined by the angular threshold size  $\theta_{thr}$ . Furthermore, it was claimed that the LGMD neuron

performs the multiplication operation and detects the angular threshold during the approach of objects along a trajectory perpendicular to the axis of the locust eye [28]. This product reaches a peak when the approaching object subtends a particular angle. This information can be used in avoiding collisions if a circuit is implemented to detect this peak-firing rate. However, this task is not easy to achieve since time is the most limited parameter in collision avoidance. In addition, the time of the peak does not really indicate the time to collision since a slower and larger object causes an earlier peak response than an object with smaller size and higher speed.

In their study, Frost et al. [77] reported that the same parameters processed by the locust are utilized by the neurons in the nucleus rotundus of pigeons in collision avoidance. However, in contrast to locusts, pigeons have neurons which signal the time to collision [77]. For the reason that pigeons have higher spatial resolution compared to those of the invertebrates, the time to collision computation can be more easily and reliably performed by their neurons.

The accurate time to collision can be calculated by using the knowledge of the distance and speed of the movement of the object. Gibson [29] introduced a parameter  $\tau$  that can signal the time to collision, where  $\tau \approx \frac{\theta(t)}{\theta'(t)}$ . Later, Lee and Reddish [51] studied diving birds and their behavioral responses. They proposed that the ratio between the retinal image size at a given instant and the rate of expansion of the image can be reliably used to compute the time to collision. Furthermore, different psychophysical experiments elicited that humans also use  $\tau$  to estimate the time to collision [50]. The problem with  $\tau$  computation is that this parameter is accurate only when the angular size of the object is not large.

As a result, a simple strategy to avoid obstacles can be achieved by building the collision sensitive LGMD model proposed by Rind et al. [63] but this model does not signal the time to collision. Therefore, a more sophisticated algorithm is required to cope with collision avoidance in complex environments. In this respect, time to collision ( $\tau$ ) computation proposed by Gibson [29] can be implemented to realize a robust and accurate collision avoidance system. In this thesis we suggest a novel hardware implementation of the time to collision computation by using the size and expansion rate parameters.

## 5.2 Hardware Architecture of the Centering Behavior

In this section, a hardware implementation of the centering behavior using a low-power analog VLSI system is described. In order to build a robust and reliable system based on this behavior, non-directional velocity information should be obtained from the optical flow field and accordingly the sums of the left and right eye motion fields should be calculated. For our hardware implementation, we utilize the Adelson-Bergen correlation-based visual motion sensor to obtain small field motion output. However, the Adelson-Bergen motion detector is not exactly a velocity sensor, rather it is a spatiotemporal frequency tuned visual motion sensor since its velocity response confounds velocity with the spatial structure of the object it is exposed to. In this context, we refer to the velocity as the ratio of the temporal and spatial frequencies, mathematically shown as follows:

$$V = \frac{w_t}{w_x} \quad (5.18)$$

where  $V$  represents velocity,  $w_t$  is the temporal frequency and  $w_x$  is the spatial frequency of the image motion.

A solution to the problem of velocity estimation is proposed by Srinivasan et al. [74]. They suggested that a velocity sensitive model can be constructed by using multiple correlators. This model can be achieved by tuning each correlator to a different spatiotemporal frequency as shown in Figure 5.4. In this way, the maximum response obtained from one of the correlators will estimate the velocity of the image. Besides, even if the contrast of the optical image changes, since all correlators will respond to this change in the same way, the correlator that produces the maximum motion output of the system will not change. This system can be easily implemented by incorporating multiple visual motion sensors with the winner-take-all circuit.

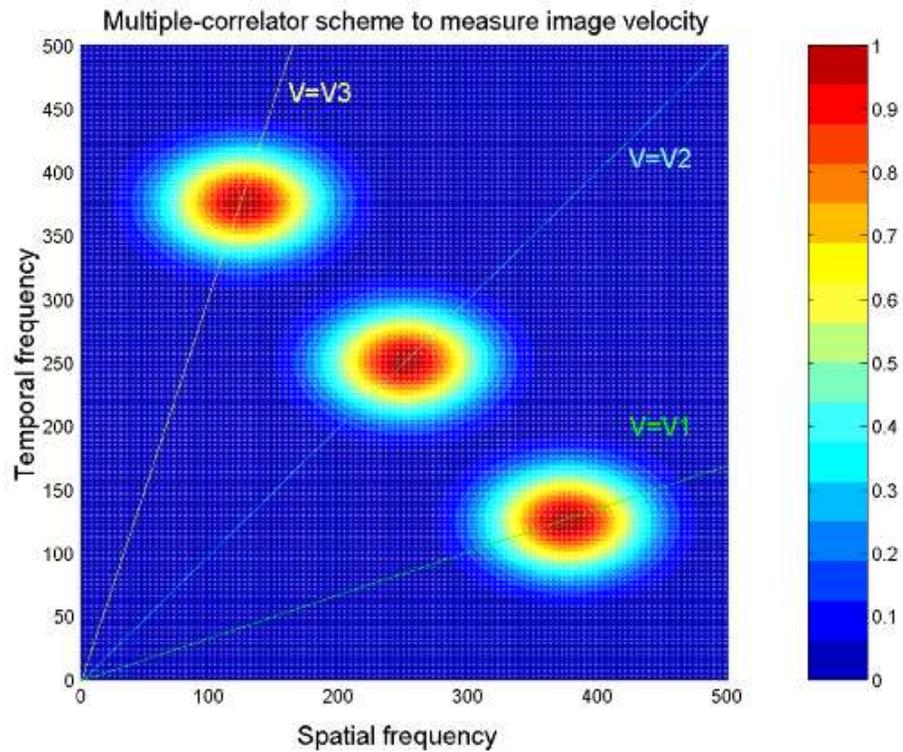


Figure 5.4: Multiple-correlator scheme realized by correlation-based motion detectors. The responses of a correlation based motion detector are shown for three different spatiotemporal frequency tuning. Each line that crosses the maximum of a correlator represents the angular velocity of the image detected by that correlator. The correlators are modelled by the two-dimensional Gaussian function. Reproduced from [74] without permission.

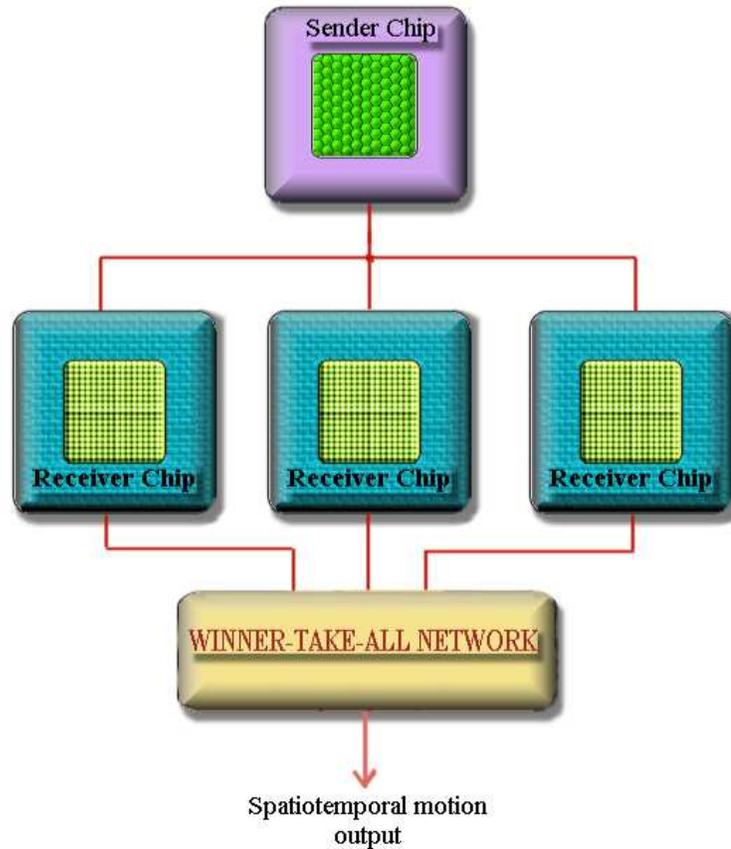


Figure 5.5: The modular implementation of the multi-correlator based velocity sensor. In this implementation one sender chip is interfaced with three receiver chips. Each receiver chip is tuned to a different spatiotemporal frequency. In this way, the winner take all network can determine the maximum response which is an estimation of optical image velocity.

A multiple-correlator based velocity sensor is computationally intensive and would require lots of transistors. Therefore, to obtain a reasonable fill factor in the front end chip, we utilized the multi-chip version of the Adelson-Bergen motion detection algorithm (see Section 4.3.2). Because the hardware implementation of the Adelson-Bergen model is space efficient and produces a highly reliable motion output, we prefer it in performing small field visual motion computation. Also, since we are interested in direction insensitive motion output we take the absolute value of small field spatiotemporal motion outputs. In Figure 5.5, an implementation is proposed to realize the multiple-correlator scheme illustrated in Figure 5.4. This system employs one sender chip and three motion computing receiver chips that are tuned to different spatiotemporal frequencies. The maximum response is determined by the winner take all circuit and estimates the velocity of optical image.

The main aim in the implementation of the centering behavior is to sum the motion outputs from the small field motion computation pixels in all receivers while keeping the velocity estimate accurate enough to realize the centering behavior. As shown in Figure 5.6, the sensed signals are transferred to three different receiver chips. Each receiver chip is tuned to a different spatiotemporal frequency. Since these receiver chips have the same high level architecture, their scanners can be synchronized to scan the same spatial positions at the same time. This enable us to compare the motion outputs of pixels in different receiver chips. The result of each comparison is stored in a

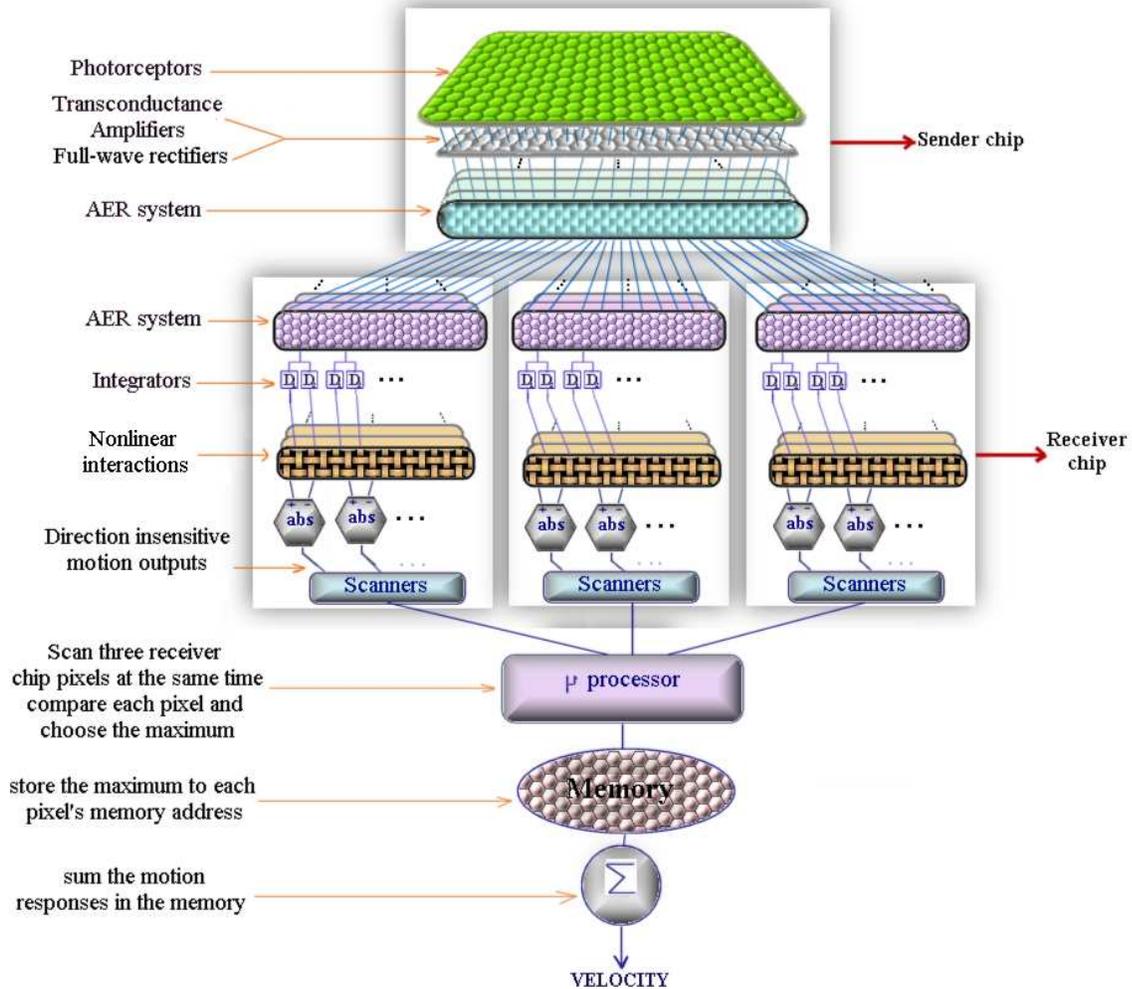


Figure 5.6: An implementation of multi-correlator based velocity sensor realized with three correlators to be used in wide field velocity detection of the retinal image. Response of the sender pixels are transmitted to three different receiver chips that are tuned to different spatiotemporal frequencies. The motion outputs of same pixels in each receiver chip are scanned by the micro processor at the same time to be able compare them. The maximum output is stored in the memory and once processor scans the last pixel, all stored motion outputs are summed.

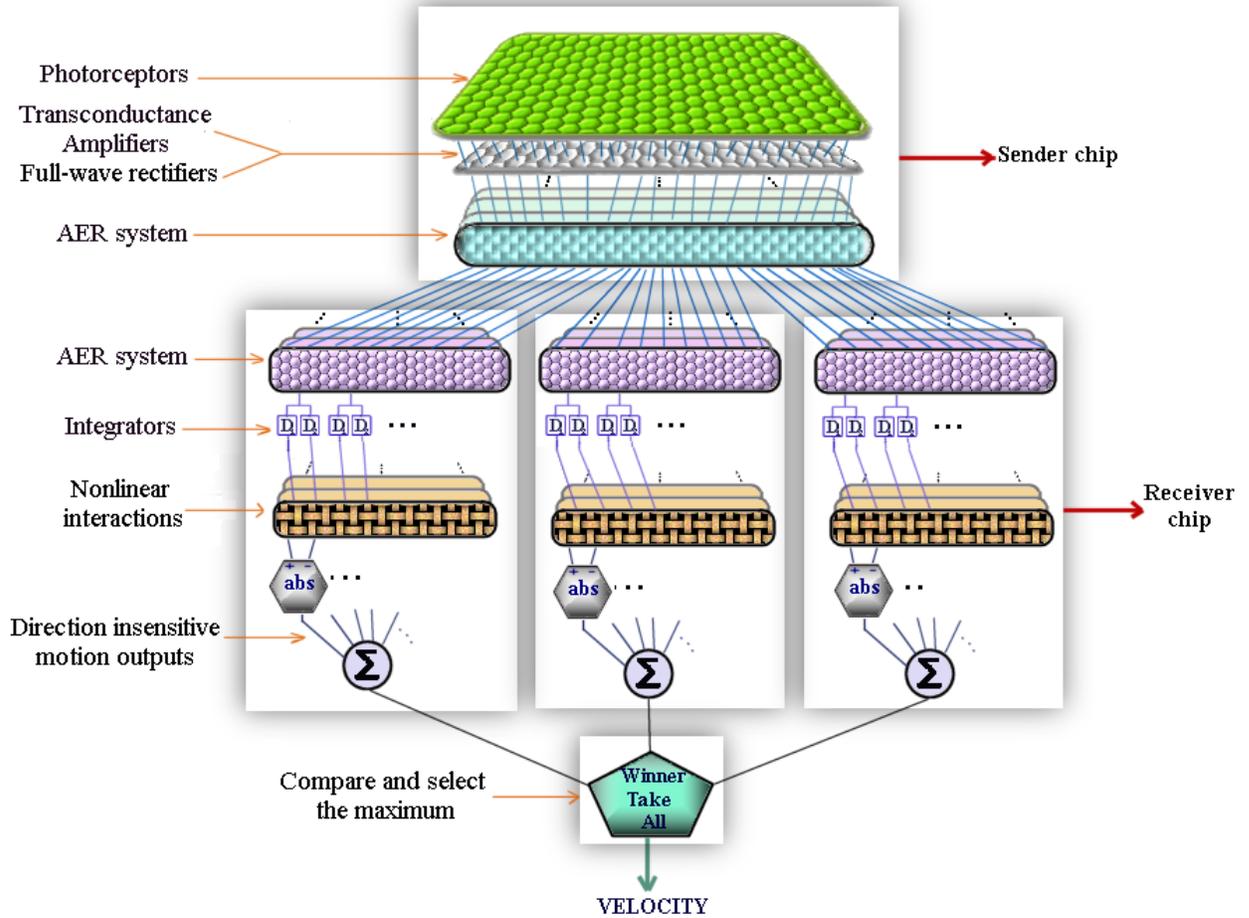


Figure 5.7: The second implementation of multi-correlator based velocity sensor. One sender and three receiver chips are employed to detect the small field motion output for three different spatiotemporal frequency tunings. The pixel responses of each receiver chip are summed individually and then compared by winner take all circuit to estimate the velocity of optical image.

memory. The memory is employed to store the maximum motion outputs of three receiver chips and to sum them whenever they are available. The sum of the maximum small field motions can be mathematically represented as follows:

$$I_{sum} = max_1(I_{EMD_1}, I_{EMD_2}, I_{EMD_3}) + max_2(I_{EMD_1}, I_{EMD_2}, I_{EMD_3}) \dots \\ + max_n(I_{EMD_1}, I_{EMD_2}, I_{EMD_3}) \quad (5.19)$$

where  $I_{EMD}$  represents the small field motion output of the elementary motion detectors and it is assumed that the system has three different types of elementary motion detectors that are each tuned to a different spatiotemporal frequency.

Apart from that, an alternative strategy can be proposed to obtain more robust results. In the computation of the centering response, we focus on the sum of the small field motion outputs to conceive the wide field properties of the optical field and because of that, small differences in the small field motion usually has small effect in determining the overall response. Therefore, instead of finding the individual maximum of the small field motions that are tuned to different spatiotemporal frequency, we can compare their overall sum to reach a decision about the velocity of the optical image. In order to implement such a computational scheme as shown in Figure 5.7, the same signal the sender chip produces can be transferred to multiple receiver chips that are tuned to different temporal frequencies. After the small field motion computation in these receiver chips, the pixel outputs are absolute valued and then summed. Accordingly, each receiver will produce a sum and this sum will be compared with other receiver outputs to decide for the maximum wide-field velocity. The computation can be formulated as follows:

$$I_{sum} = max(sum_1(EMD_1, \dots, EMD_n), sum_2(EMD_1, \dots, EMD_n), \\ sum_3(EMD_1, \dots, EMD_n)) \quad (5.20)$$

In this formula, we assume that there are three different receiver chips tuned to different spatiotemporal frequencies. This kind of computation is not equivalent to summing the individual maximum of the small field motion, however, in this way, we can decrease the complexity and the computational load of the overall system.

### 5.2.1 Circuit implementation of the direction insensitive movement detector neuron

In this section, we explain the circuit implementation of the direction insensitive motion detection neuron that mediates the centering behavior. We realize this neuron by utilizing a low-power VLSI multi-chip system.

A multi-chip system can be implemented by incorporating a sender chip with one or multiple receiver chips. Each pixel in the sender has an AER circuit so that immediate changes can be reliably sent to receiver chips. Also, the amplitude of the changes sensed by each photoreceptor is encoded by the firing rate of the AER system so that within a small time frame the response of a photoreceptor can be transferred to the receivers.

A receiver chip is utilized to perform small- and wide-field motion computations. It employs the elementary motion detectors (EMD) to obtain small field motion vectors. For that purpose Adelson-Bergen motion detectors (see Section 4.3.2) are used and the outputs of these detectors are absolute valued to realize the direction-insensitive visual motion sensor. Each detector in the receiver chip integrates the signals sent by the sender and performs the small-field motion computation. These pixels integrate the signals from the sender so that the amplitude of the changes in the intensity level of the image sensed by each photoreceptor can be reconstructed and necessary delays to realize the correlation scheme can be obtained (see Chapter 4).

The P and N pixels that integrate the positive and negative parts of the signal sent by the sender

are illustrated in Figure 5.8a and Figure 5.8b, respectively. The P pixel is designed to contain the motion computation circuitry, whereas the N pixel includes only AER interface circuit and integrators. After the integration stage, the motion computation is performed by using subtraction and absolute value circuits. The current output of the Adelson-Bergen sensor can be formulated as follows:

$$I_{motion} = |I_{2negD} + I_{1neg} - I_{2posD} - I_{1pos}| + |I_{2neg} + I_{1posD} - I_{2pos} - I_{1negD}| \\ - |I_{2posD} + I_{1neg} - I_{2negD} - I_{1pos}| - |I_{2neg} + I_{1negD} - I_{2pos} - I_{1posD}| \quad (5.21)$$

After this stage, an absolute value circuit is added to obtain the final direction-insensitive motion output. The final motion output of each EMD can be formulated as follows:

$$I_{output} = |I_{motion}| \quad (5.22)$$

The wide-field motion output is obtained by making use of the capabilities of the scanners that help us sum the motion outputs of all EMDs. In this way, we obtain a sender-receiver chip pair that can qualitatively signal the motion output created by its own motion or movements of objects in its environment.

This system can be used to detect direction-insensitive motion sensed by the right and left eyes. The motion output does not represent the motion velocity but an implementation of the multi-correlator based velocity sensor is straightforward with a modular design strategy.

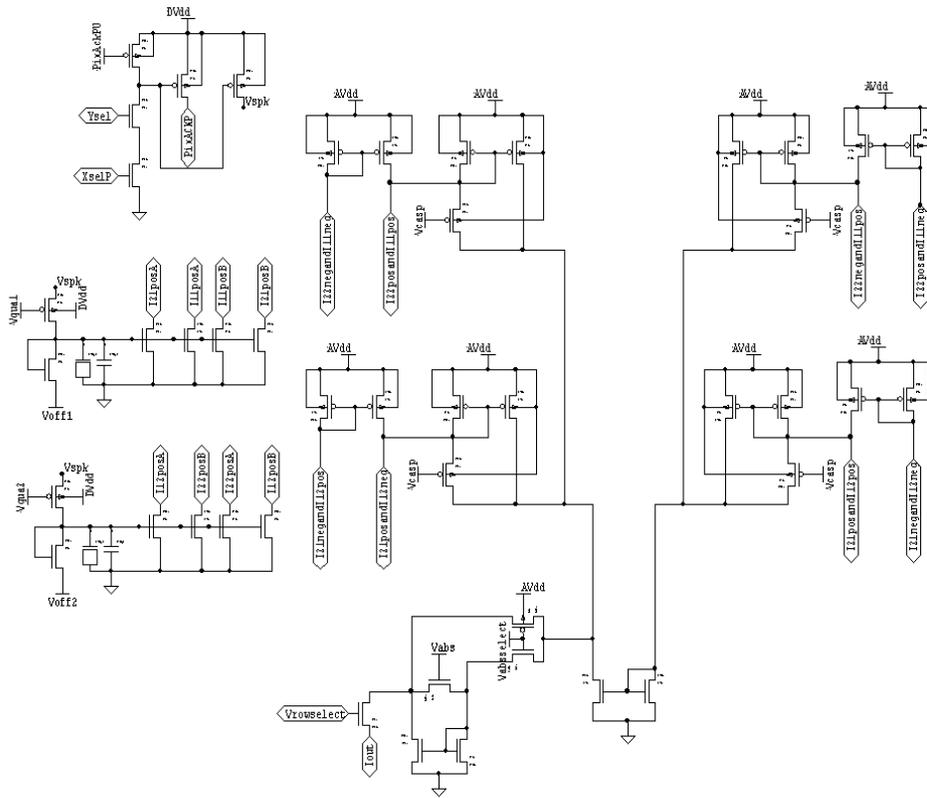
If a multi-correlator based velocity sensor is desired, then receiver chips that are tuned to different spatiotemporal frequencies can be achieved by using the capabilities of the AER circuitry. Firstly, the temporal frequency tuning of each chip can be adjusted by changing the biases of the integrator circuit. Secondly, the spatial tuning can be changed by interchanging the columns of the AER system in such a way that instead of computing the motion response of neighboring pixels, interaction between odd-numbered pixels or even-numbered pixels is used. This strategy can be achieved by changing the column numbers of the receiver chip as illustrated below:

Sender chip	Receiver chip
1	1
2	3
3	5
4	2
5	4
6	6

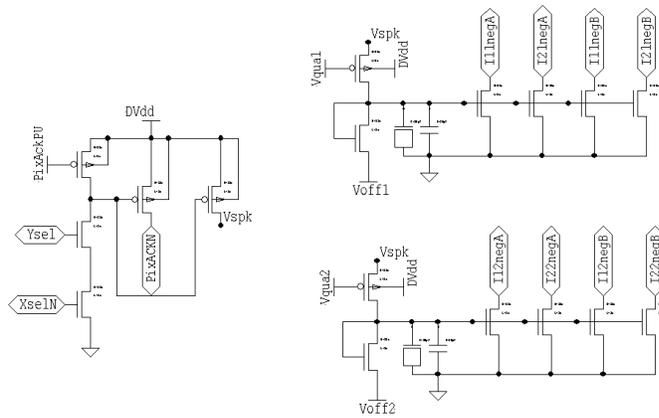
The unwanted interaction between the second and fifth columns can be avoided by implementing the circuit in such a way that there will not be an interaction between these columns. Although this strategy is appealing in terms of its simplicity, the cost of building extra receiver chips makes this implementation unattractive. For that purpose, a PIC can be employed to map the addresses sent from the sender to the receiver chip. Besides, since in this case we do not have any control over the internal structure of the chip, we need to use a new mapping strategy to avoid unrelated interaction between the second and fifth columns. An alternative mapping can be realized as follows:

Sender chip	Receiver chip
1	1
2	3
3	5
4	6
5	4
6	2

In this case we will have motion output obtained from the interaction between the fifth and sixth



(a)



(b)

Figure 5.8: The implementation of the circuit that integrates the positive and negative parts of the signal sent from the sender and computes the final motion output. (a) The P pixel that consist of AER interface circuit, integrators, absolute value circuits and subtractors. (b) The N pixel that only integrates the negative part of the signals.

columns of the sender. This non-direction selective velocity sensor system mediates the centering behavior. It can be used in applications where an agent is needed to navigate between objects in a cluttered environment.

The final chip contains  $6 \times 6$  pixels and the layouts of the pixel and the chip are shown in Figure A.18 and Figure A.19, respectively.

### 5.3 Hardware Architecture of the Escape Behavior

Although the centering behavior is a very efficient way of dealing with navigation in cluttered environments, it becomes ineffective when it is challenged by objects in the heading direction. Particularly, the symmetric expansion of an object in the heading direction will not cause a robot driven by the centering response to turn away. For this reason, a looming sensitive sensor should be implemented and placed in the heading direction of the system. In this way, during the pure translatory movements of the robot, the looming sensitive sensor can signal the objects that are on a collision course or can compute the time to collision parameter so that the angle of rotation to escape from the object can be calculated without initiating a sudden turn.

Time to collision is defined as  $\tau = \frac{\theta}{\frac{\partial \theta}{\partial t}}$  and in Equation 5.9 it is found as  $t_{collision} = \frac{m}{V_r}$ . In 1D motion computation, if the object is a rigid body without texture, this formula can be implemented by detecting the symmetric (or close to symmetric) movement of the maximum velocity vector in the visual field of each eye. In order to analyze the wide-field motion in the heading direction, we separated the visual field into left and right parts. As in the case of the implementation of the centering behavior, we utilized the multiple-correlator velocity model. The reason for this choice is that we need to find the expansion rate and this can be achieved only by finding the velocity of the edges of the object. We are particularly interested in the right motion in the right eye and the left motion in the left eye because the looming state of an object can be detected from its size and expansion rate. Such a system can be implemented by using the right sensitive motion detectors in the right eye and the left motion sensitive motion detectors in the left eye.

Firstly, we assume a symmetric or quasi-symmetric expansion of an object in the heading direction, because in other cases, the centering response can initiate enough turn to escape from the object. Therefore, the real problem is to be able to signal an escape response when the object is looming symmetrically on a collision course. Secondly, we assume that  $\theta$  or the size of the object can be estimated by incorporating the maximum operation on the velocity computation pixels. This helps us detect the maximum velocity that identifies the closest part of the object and/or its edges in the visual field of each eye. In our algorithm, the maximum operation is improved by employing a lateral inhibition and excitation network.

Looming is a real time response to movements on a collision course and therefore it has to be tracked and integrated in time. For this reason, after finding the maximum velocity on each eye, they need to be tracked in time and checked if the rightward maximum velocity in the right eye is moving to the right and if the leftward maximum velocity in the left eye is moving to the left. In the case of expansion, the position changes of the rightward and leftward maximum velocities include a few expansion parameters such as the time-to-collision and the degree of symmetric expansion. Also, the correlation between the positions of the left and right maximum velocities can be found to obtain the information on the object's degree of symmetric expansion. Assuming that the center of the visual field is the origin of the position vector, then we can relate the positions of the maximum velocities to find the symmetry parameter. For instance, if the magnitude of the maximum velocity positions on each eye are very different, then this information gives us a clue that the expansion is not symmetric and the centering behavior has to be employed to avoid collisions in such cases. Therefore, the symmetry parameter can be integrated with the final output to limit the cases when the escape behavior algorithm does not respond to collisions.

An alternative strategy can be employed by changing the maximum operation with a group of velocities limited by the maximum velocity operation. This operation finds the maximum velocity in the motion field and looks for other velocities that are more than a constant fraction of the maximum velocity (where this fraction is defined in the interval  $0 < fraction < 1$ ). Therefore, we determine

the threshold velocity that can be used to find the velocities of interest. The main purpose of this operation is to track the closest part of the object and from this information to find the size and expansion rate. In addition, the rate of expansion can be computed by averaging the velocities that are bigger than the computed threshold. This can be achieved by determining the number of velocities bigger than threshold and dividing the sum of velocities with this number.

As a result of these observations, we can find the degree of looming by utilizing the formula found in Equation 5.9 and formulate the escape parameters that initiate the escape behavior as follows:

$$I_{timeleft} = \frac{\theta}{\frac{\delta\theta}{\delta t}} \quad (5.23)$$

$$I_{symmetry} = \frac{x_{maxright} - x_{maxleft}}{x_{maxright} + x_{maxleft}} \quad (5.24)$$

$$looming = \frac{1}{I_{timeleft}} = \frac{\frac{\delta\theta}{\delta t}}{\theta} \quad (5.25)$$

where  $x$  is the sum of the positions of the velocities bigger than threshold. The value of the looming signals the degree of danger and therefore provides a better indication about possible collisions than the time-to-collision parameter does. The algorithm that illustrates the computation performed to find these parameters is shown in Figure 5.9. This algorithm is constructed by employing the multi-correlator velocity sensor method. In contrast to other algorithms, the visual field is divided into two parts that represent the left and right eyes. Accordingly, the motion output is obtained by acquiring the positive part of the right eye motion and the negative part of the left eye motion (assuming preferred motion is in the rightward direction). After that, the maximum motion output in each eye's visual field is determined by the winner-take-all circuit. After that a fraction of the maximum motion output is used to threshold the small field motion responses on each eye. The positions and values of the motion outputs that are bigger than the threshold are summed to be used in the calculation of the expansion parameters. Firstly, the velocity of each eye is determined by utilizing the winner-take-all circuit as a part of the multi-correlator model. Secondly, the symmetry of expansion is determined by the contrast calculation of the left-right eye motions of the correlator enabled by the winner-take-all circuit. Finally, the looming parameters can be used to perform the time-to-collision computation and to find the degree of danger by integrating the response of these computations in time.

## 5.4 Algorithm Implemented on an Agent

In this section, we explain a system application of the centering and escape behaviors incorporated onto an agent. This agent is illustrated in Figure 5.10.

The visual field is divided into five regions. The sender chips are used as front-end chips and responsible for signalling the illumination changes in their visual field. Two sender chips on each side are employed in order to increase the visual field of the agent. The small field motion computation is performed by the receiver chips. After obtaining the optical flow field, outputs of the receiver chips are used in the wide-field motion computation performed by the comparison circuit, sum circuit and micro-processors.

As is indicated in the implementations of the centering and escape behaviors, the obstacle avoidance system incorporates the multi-correlator velocity algorithm and therefore for each sender, three receiver chips are employed. The comparison circuit finds the maximum sum of the velocities computed by the receiver chips and in this way yields an estimate of the wide-field velocity.

In order to implement the centering behavior, the responses of the comparison circuits are summed by using a sum circuit. As a result of this operation, we can get an estimate of the rightward and leftward motion. The responses are sent to the micro-processor which processes this

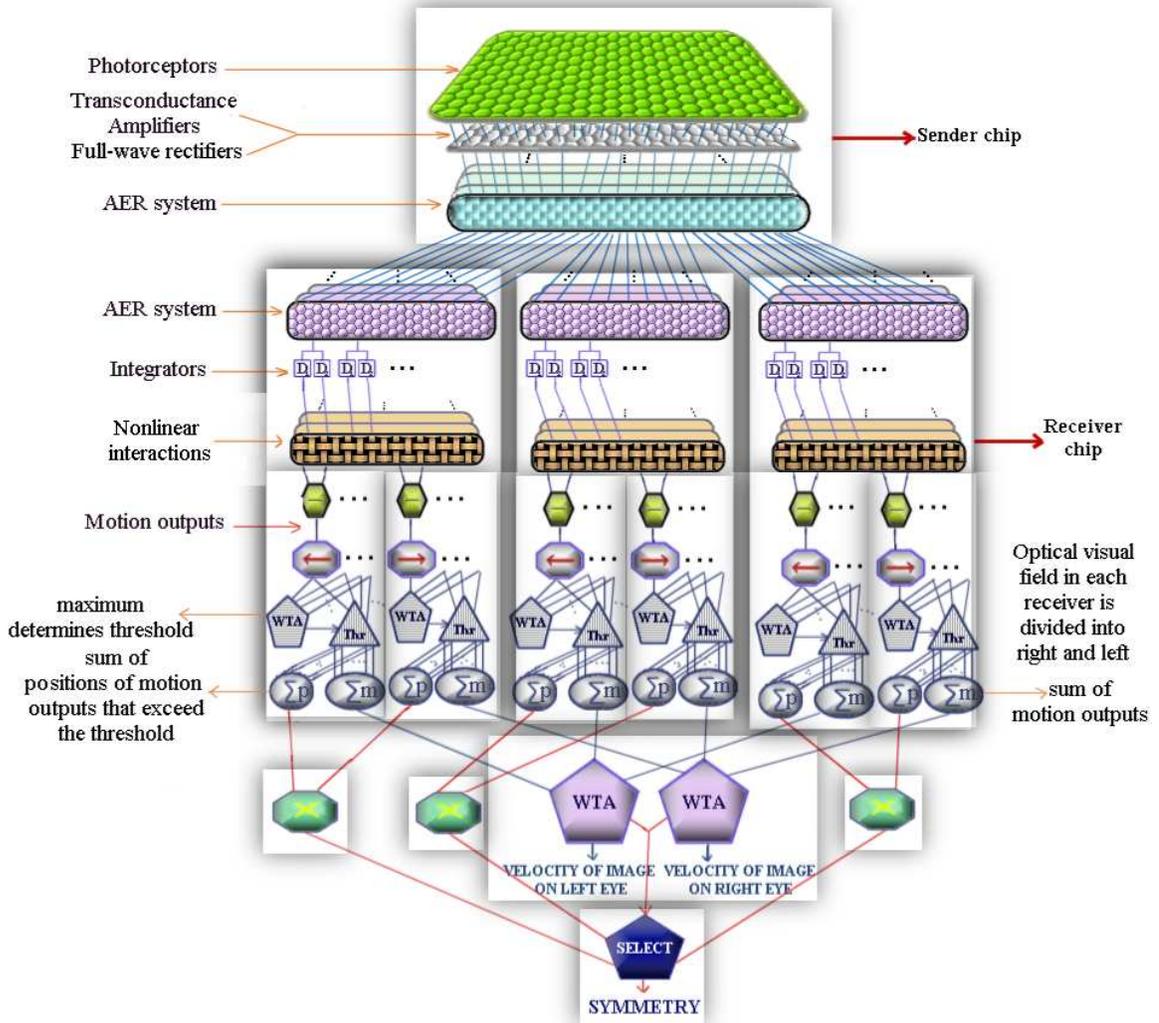


Figure 5.9: The final implementation of the escape response. One sender and three receiver chips are employed to extract the small field motion output. Each receiver chip is tuned to a different spatiotemporal frequency and its optical field is divided into two regions to represent the right and left eyes. The motion outputs on each eye are rectified and then the maximum rectified output on each eye becomes the threshold of motion responses. After the threshold stage, the sum of the positions and magnitudes of motion signals are calculated to calculate the symmetry, and the right and left eyes motions.

information and mediates the movement of the agent. The process includes finding the contrast of the right motion relative to the left motion. Accordingly, contrast levels that are close to zero inhibit rotation and allow only translatory movement of the agent. Whereas, motion contrasts that are close to plus or minus one create the largest rotatory movements.

The centering behavior is utilized to find the safest way in cluttered environments. Also, we incorporated the escape behavior which detects collisions in the heading direction to improve the capabilities of the agent. Similar to the implementation of the centering behavior, escape behavior is realized by using a sender and three receiver chips. However, unlike the receiver chips used in centering behavior, these receiver chips include the motion detectors that extract the rightward motion in the right part and the leftward motion in the left part of the visual field. Moreover, maximum and threshold operations are employed in the chips to find the velocities bigger than some fraction of the maximum velocity on each part of the visual field. As a result of this computation we obtain motion outputs and their positions from the left and right parts of each receiver chip. These motion computation modules are integrated with a comparison circuit and micro-processor. The comparison circuit finds the maximum velocity pair for the left and right motion and enables the sum of the positions of motion outputs. After that, the micro-processor takes these motion output and position values and processes them to find the expansion parameters. In the case of a predetermined symmetry value the tracking of motion output positions can be enabled by the micro-processor. Therefore, the position change, which signals the change in the size of the object, and the average of the velocity values are combined to find the degree of looming caused by the motion of the agent. Looming is a continuous time operation. For this reason, after each discrete movement of the agent, the looming value has to be stored in a memory. The time frame with its associated threshold looming value can be used to determine if the integrated looming value is signaling any danger.

This agent with its escape and centering behavior can allow successful navigation in cluttered and hazardous environments and is therefore well-suited for robotics applications.

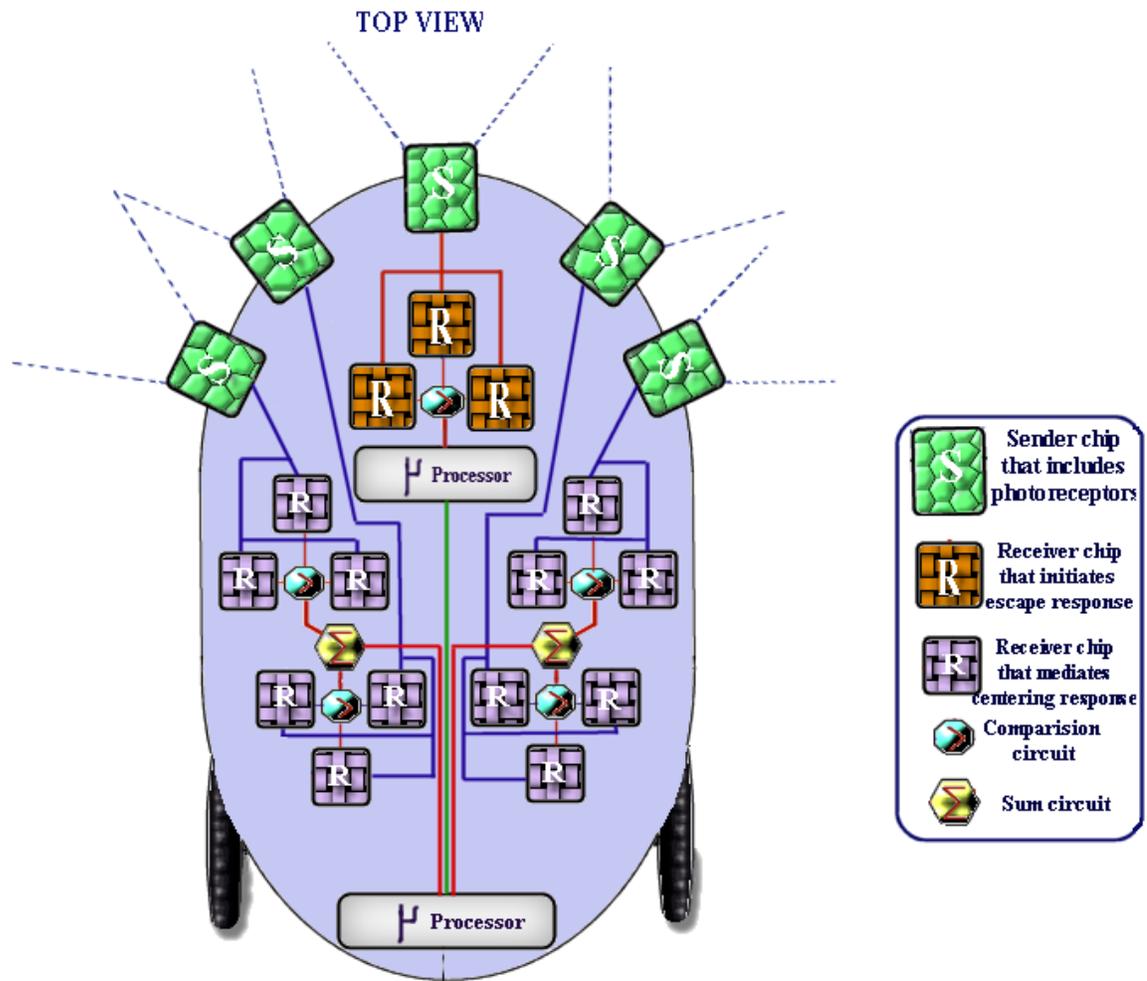


Figure 5.10: The final implementation of the centering response and the escape response on an agent. 5 sender chips are used to increase the visual angle of the agent. Receiver chips extract the small field motion outputs and process them. Comparison circuits, sum circuits and microprocessors are responsible for escape response and centering behavior.

## Chapter 6

# Discussion

In this thesis, monolithic and multi-chip implementations of neuromorphic visual motion models were explained and a biologically inspired obstacle avoidance system based on the anatomy and physiological behaviors of flying insects was described.

Firstly, we presented a monolithic implementation of the hysteretic winner-take-all based motion sensor. This motion sensor exploits the inhibition mechanism that is inherent in the winner-take-all circuit. Barlow and Levick described this mechanism to explain the direction selectivity in the rabbit retina. However, we did not directly implement their model, rather, we used the same mechanism to implement a novel algorithm. This algorithm yields a compact and reliable motion detection model and makes it possible to realize a high fill factor in the sensors. Although it is very efficient in motion detection, this motion sensor has its own limitations. The threshold that is used to determine the winner of two input signals may cause problems in noisy environments. Since it sets the winner state to first signal that exceeds it, any noise in the system or environment prevents the sensor from functioning reliably. In order to overcome the effect of noise, the threshold level should be increased. However, in this case, the response of the sensor to low-contrast stimuli vanishes.

Secondly, we described another monolithic visual motion sensor that is based on the nonlinear differentiator. As is the nature of differentiation, this model amplifies the noise in the environment. In order to decrease this effect, we exploited the adaptive characteristic of the photoreceptor to obtain a temporal window. In addition, we used a differential amplifier before the nonlinear differentiation stage. This circuit adds a condition to the response of the circuitry. If the DC level of one of the photoreceptor outputs is higher than the neighboring photoreceptor output, then the noise that is overridden onto these signals may not effect the efficiency of the system. In turn, this helps the differentiator stay silent even in the presence of noise. However, if the DC level of the photoreceptors are the same, then the differentiator amplifies the noise should its amplitude be big enough. Otherwise, the multiplication of this signal with the temporal window helps the system depress the noise.

Furthermore, we have described and characterized intensity based multi-chip analog VLSI motion sensors and presented a new technique to realize the temporal filters needed for motion computation. By employing a modular strategy, we implemented the Hassenstein-Reichardt correlation algorithm, the Adelson-Bergen spatiotemporal energy model and the Barlow-Levick motion detection models. The characterization results clearly elucidate the fact that by using this technique, we can obtain a reliable, low-power and real time multi-chip neuromorphic motion processing system while retaining many of the advantages of monolithic implementation.

These systems were implemented by using a sender and receiver chip pair. The sender chip was employed to transfer the intensity changes to the receiver chip that has the motion computation circuitry. The sender chip does not contain any interaction between pixels. In turn, this helped to improve the fill factor of the sender chip. In this chip we were able to obtain  $6 \times 6$  pixels in a standard  $1.5\mu\text{m}$  CMOS process and  $2.1\text{mm} \times 2.1\text{mm}$  die size. In this study, we presented a new design that eliminates one of the two AER interface circuits in the sender pixel and increases the fill factor of the chip. This sender chip will be used as a new front-end chip and we are expecting to

exceed the number of pixels that can currently be realized on a single chip.

In the implementation of the receiver chips that have motion computation circuitry, we obtained the best results in terms of space from the Barlow-Levick algorithm. The circuit implementation of the Hassenstein-Reichardt algorithm in the receiver chip is achieved with 65 transistors and 4 capacitors. The implementation of the Adelson-Bergen algorithm in the receiver includes 55 transistors and 4 capacitors in each pixel. Lastly, the Barlow-Levick algorithm was implemented by using 34 transistors and 4 capacitors. The Barlow-Levick algorithm yields the most compact solution to the motion computation.

The characterization of the sensors showed that these intensity-based sensors exhibit very similar spatiotemporal characteristics. Therefore, there is not much difference in their performances. In this case, the number of transistors used becomes more important and the Barlow-Levick implementation provides the most space efficient multi-chip system. The characterization results of these multi-chip sensors revealed that they respond to the optimal spatial frequencies over a velocity range of more than an order of magnitude. These sensors are tuned to spatiotemporal frequencies. Hence, they were implemented in a way that the range of the spatial and temporal frequencies and the contrast level to which they can respond are maximized. Furthermore, 2D motion computation can be easily achieved with this implementation by using a single sender chip with two receivers and manipulating the  $x$  and  $y$  connections of the address lines of the second receiver. These multi-chip biomimetic vision sensors will allow complex visual motion computations to be performed in real-time.

Lastly, we designed a biomimetic obstacle avoidance system by exploiting the centering and escape behaviors observed in flying insects. In the motion computation, we utilized motion parallax to obtain information on the obstacles and structure of the environment. In order to achieve navigation in cluttered environments, we implemented the centering behavior which is mediated by a direction insensitive movement detecting system. By taking the absolute value of the Adelson-Bergen algorithm output, we suggested an implementation of a multi-chip nondirection selective motion sensor to realize the centering behavior. In this thesis, we proposed that multi-correlator scheme can be used to improve the reliability of the velocity output obtained from the implemented sensor. The main problem with the centering behavior is that it is not that effective in avoiding collisions with objects that are in the heading direction. We tried to solve this problem by utilizing the escape response which is initiated by the lobula giant movement detector in locusts to avoid predators. Together with the centering behavior we obtained profoundly promising results from the simulations. However, the proposed system in Figure 5.10 has to be improved in terms of the number of motion chips used in order to make the system more practical in robotics applications. Furthermore, the optimization has to be done to decrease the number of sender chips and to increase the angle of the visual field.

# Bibliography

- [1] Indiveri, G. and Liu, S.C., Lecture notes of the course “Computation in Neromorphic Analog VLSI Systems” given in Institute of Neuroinformatics at University of Zurich.
- [2] E.H. Adelson and J.R. Bergen, *Spatiotemporal energy models for the perception of motion*, J. Optical Society of America A **2** (1985), no. 2, 284–299.
- [3] P. Anandan, *A computational framework and an algorithm for the measurement of visual motion*, International Journal of Computer Vision **2** (1989), 283–310.
- [4] A.G. Andreou and K. Strohbehn, *Analog VLSI implementation of the Hassenstein-Reichardt-Poggio model for vision computation*, Proc. IEEE Intl. Conf. on Systems, Man, and Cybernetics (Los Angeles, CA), November 1990, pp. 708–710.
- [5] H.B. Barlow, M.R. Hill, and W.R. Levick, *Retinal ganglion cells responding selectively to direction and speed of image motion in the rabbit*, J. Physiol. **173** (1964), 377–407.
- [6] H.B. Barlow and W.R. Levick, *The mechanism of directionally selective units in the rabbit’s retina*, Journal of Physiology **178** (1965), 447–507.
- [7] R.G. Benson and T. Delbrück, *Direction selective silicon retina that uses null inhibition*, Advances in Neural Information Processing Systems 4 (San Mateo, CA) (D.S. Touretzky, ed.), Morgan Kaufmann, 1991, pp. 756–763.
- [8] M. Blanchard, F.C. Rind, and P. F. M. J. Verschure, *Collision avoidance using a model of the locust LGMD neuron*, Robotics and Autonomous Systems **30** (2000), 17–38.
- [9] K. Boahen, *NSF neuromorphic engineering workshop report*, Telluride, CO, 1996.
- [10] A. Borst, *Models of motion detection*, Nature Neuroscience **3** (2000), 1168.
- [11] A. Borst and M. Egelhaaf, *Principles of visual motion detection*, Trends in Neuroscience **12** (1989).
- [12] K. Bult and H. Wallinga, *A class of analog CMOS circuits based on the square-law characteristic of a MOS transistor in saturation*, IEEE Journal of Solid-State Circuits **SC-22** (1987), no. 3, 357–365.
- [13] C.C. Chang and S.I. Liu, *Weak inversion four quadrant multiplier and two-quadrant divider*, Electronic letters **34** (1998).
- [14] T. Delbrück and C.A. Mead, *Analog VLSI phototransduction*, Tech. Report 30, Computation and Neural Systems Program, California Institute of Technology, 1993.
- [15] R. Deutschmann, C.M. Higgins, and C. Koch, *Real-time analog VLSI sensors for 2-D direction of motion*, Proceedings of the Int. Conf. on Artificial Neural Networks, Springer Verlag, 1997, pp. 1163–1168.

- [16] R. A. Deutschmann and C. Koch, *Compact real-time 2-D gradient based analog VLSI motion sensor*, Proceedings of the Int. Conf. on Advanced Focal Plane Arrays and Electronic Cameras (Zurich/Switzerland), 1998.
- [17] S.P. DeWeerth and T.G. Morris, *Analog VLSI circuits for primitive sensory attention*, Proc. IEEE Int. Symp. Circuits and Systems, vol. 6, IEEE, 1994, pp. 507–510.
- [18] ———, *CMOS current mode winner-take-all circuit with distributed hysteresis*, Electronics Letters **31** (1995), no. 13, 1051–1053.
- [19] R.O. Dror, D.C. O’Carroll, and S.B. Laughlin, *Accuracy of velocity estimation by Reichardt correlators*.
- [20] A.P. Duchon and W.H. Warren, *Robot navigation from a Gibsonian viewpoint*, IEEE Trans. Syst. Man Cybern. (1994), 2272–2277.
- [21] S. Edelman, *Constraints on the nature of the neural representation of the visual world*, Trends in Cognitive Sciences **6** (2002), 125–131.
- [22] M. Egelhaaf, K. Hausen, W. Reichardt, and C. Wehrhahn, *Visual course control in flies relies on neuronal computation of object and background motion*, Trends in Neuroscience **11** (1988), 351–358.
- [23] M. Egelhaaf and R. Kern, *Vision in flying insects*, Current Opinion in Neurobiology **12** (2002), 699–706.
- [24] R. Etienne-Cummings, J. Van der Spiegel, and P. Mueller, *A focal plane visual motion measurement sensor*, IEEE Trans. on Circuit and Systems I **44** (1997), no. 1, 55–66.
- [25] ———, *Hardware implementation of a visual-motion pixel using oriented spatiotemporal neural filters*, IEEE Transactions on Circuits and Systems-II **46** (1999), no. 9, 1121–1136.
- [26] C.L. Fennema and W.B. Thompson, *Velocity determination in scenes containing several moving object*.
- [27] N. Franceschini, J.M. Pichon, and C. Blanes, *From insect vision to robot vision*, Philosophical Transactions of the Royal Society of London B **337** (1992), 283–294.
- [28] F. Gabbiani, H. Krapp, and G. Laurent, *Computation of object approach by a wide-field, motion sensitive neuron*, J. Neuroscience **19** (1999).
- [29] J. Gibon, J., *The ecological approach to visual perception*, Boston, MA: Houghton Mifflin, 1979.
- [30] R.R. Harrison and C. Koch, *An analog VLSI model of the fly elementary motion detector*, Advances in Neural Information Processing Systems 10 (M. I. Jordan, M. J. Kearns, and S. A. Solla, eds.), MIT Press, 1998, pp. 880–886.
- [31] B. Hassenstein and W. Reichardt, *Systemtheoretische analyse der Zeit-, Reihenfolgen- und Vorzeichenauswertung bei der Bewegungsperzeption des Rüsselkäfers Chlorophanus*, Zeitschrift für Naturforschung **11b** (1956), 513–524.
- [32] N. Hatsopoulos, F. Gabbiani, and G. Laurent, *Elementary computation of object approach by a wide-field visual neuron*, **270** (1995), 1000–1003.
- [33] D.J. Heeger, E.P. Simoncelli, and J.A. Movshon, *Computational models of cortical visual processing*, Proc. Natl. Acad. Sci. **93** (1996), 623–627.
- [34] C.M. Higgins, R. A. Deutschmann, and C. Koch, *Pulse-based 2D motion sensors*, IEEE Transactions on Circuits and Systems II **46** (1999), no. 6, 677–687.

- [35] C.M. Higgins and C. Koch, *Multi-chip neuromorphic motion processing*, Proc. of the 20th Conference on Advanced Research in VLSI (Atlanta, GA), 1999.
- [36] ———, *A modular multi-chip neuromorphic architecture for real-time visual motion processing*, Analog Integrated Circuits and Signal Processing **24** (2000), no. 3.
- [37] C.M. Higgins and S.K. Korrapati, *An analog VLSI motion energy sensor based on the Adelson-Bergen algorithm*, Proceedings of the International Symposium on Biologically-Inspired Systems, December 2000.
- [38] T. Horiuchi, W. Bair, B. Bishofberger, J. Lazzaro, and C. Koch, *Computing motion using analog VLSI chips: an experimental comparison among different approaches*, International Journal of Computer Vision **8** (1992), 203–216.
- [39] S.A. Huber and H.H. Bülthoff, *Visuomotor control in flies and behavior-based agents*, Biological Inspired Robot Behavior Engineering, Studies in Fuzziness and Soft Computing (J. Santos R. J. Duro and M. Granas, eds.), Physica-Verlag, Heidelberg, 2002, pp. 89–117.
- [40] G. Indiveri, *Analogue VLSI model of locust DCMD neuron response for computation of object approach.*, Neuromorphic Systems: engineering silicon from neurobiology (L.S. Smith and A. Hamilto, eds.), 1998, pp. 47–60.
- [41] ———, *Neuromorphic analog VLSI sensor for visual tracking: Circuits and application examples*, (1999).
- [42] G. Indiveri and G.M. Bisio, *Analog subthreshold VLSI implementation of a neuromorphic model of the visual cortex for pre-attentive vision*, 1994.
- [43] G. Indiveri, J. Kramer, and C. Koch, *Parallel analog VLSI architectures for computation of heading direction and time-to-contact*, Advances in Neural Information Processing Systems (Cambridge, MA) (D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo, eds.), vol. 8, MIT, 1996, pp. 720–726.
- [44] Boahen K., *Retinomorphich chips that see quadruple images*, Proceedings of the 7th International Conference on Microelectronics for Neural, Fuzzy and Bio-inspired Systems, April 1999.
- [45] ———, *A throughput-on-demand 2-d address-event transmitter for neuromorphic chips*, Proc. of the 20th Conference on Advanced Research in VLSI (Atlanta, GA), 1999.
- [46] J. Kramer, *Compact integrated motion sensor with three-pixel interaction*, IEEE Trans. Pattern Analysis and Machine Intelligence **18** (1996), 455–460.
- [47] J. Kramer, R. Sarpeshkar, and C. Koch, *An analog VLSI velocity sensor*, Proc. Int. Symp. Circuit and Systems (ISCAS) (Seattle, WA), May 1995, pp. 413–416.
- [48] ———, *Pulse-based analog VLSI velocity sensors*, IEEE Trans. Circuits and Systems II **44** (1997), 86–101.
- [49] J. Lazzaro, S. Ryckebusch, M.A. Mahowald, and C.A. Mead, *Winner-take-all networks of  $O(n)$  complexity*, Advances in neural information processing systems (San Mateo - CA) (D.S. Touretzky, ed.), vol. 2, Morgan Kaufmann, 1989, pp. 703–711.
- [50] D. N. Lee, *A theory of visual control of braking based on information about time-to-collision*, Perception **5** (1976), 437–459.
- [51] D.N. Lee and P.E. Reddish, *Plummeting gannets: a paradigm of ecological optics*, Nature **293** (1981), 293–294.
- [52] J.Y. Lettvin, H.R. Maturana, W.S. McCulloch, and W.H. Pitts, *What the frog's eye tells the frog's brain*, Proceedings of the Institute of Radio Engineers **47** (1959), 1940–1951.

- [53] M.A. Lewis, *Visual navigation in a robot using zigzag behaviour*, The MIT Press, Cambridge, 1998.
- [54] J.O. Limp and J.A. Murphy, *Estimating the velocity of moving images in television signals*.
- [55] S.C. Liu, *Neuromorphic models of visual and motion processing in the fly visual system*, Ph.D. thesis, Department of Computation and Neural Systems, California Institute of Technology, Pasadena, CA., 1997.
- [56] M.A. Mahowald, *VLSI analogs of neuronal visual processing: a synthesis of form and function*, Ph.D. thesis, Department of Computation and Neural Systems, California Institute of Technology, Pasadena, CA., 1992.
- [57] B.A. McGuire, C.D. Gilbert, P.K. Rivlin, and T.N. Wiesel, *Targets of horizontal connections in macaque primary visual cortex*, *J. Comp. Neurology* **305** (1991), 370–392.
- [58] C.A. Mead, *Neuromorphic electronic systems*, *Proceedings of the IEEE* **78** (1990), 1629–1636.
- [59] F Mura and N Franceschini, *Visual control of altitude and speed in a flying agent*, *Proceedings of the 3th International Conference on Simulation of Adaptive Behaviour* (D Cliff, P Husbands, JA Meyer, and Wilson SW, eds.), 1994, pp. 91–99.
- [60] S.J. Nowlan and T.J. Sejnowski, *Filter selection model for motion segmentation and velocity integration*, *J. Optical Soc. America* **11** (1994), no. 12, 3177–3200.
- [61] N. Qian, RA Andersen, and EH Adelson, *Transparent motion perception as detection of unbalanced motion signals. III. modeling*, *J. Neuroscience* **14** (1994), no. 12, 7381–7392.
- [62] W. Reichardt, *Autocorrelation, a principle for the evaluation of sensory information by the central nervous system*, *Sensory Communication* (WA Rosenblith, ed.), MIT Press, New York, 1961, pp. 303–317.
- [63] F.C. Rind and P.J. Simmons, *Neural network based on the input organization of an identified neuron signaling impending collision*, *J Neurophysiol* **75** (1996), 967–985.
- [64] \_\_\_\_\_, *Signaling of object approach by the dcmd neuron of the locust*, *J Neurophysiol* **77** (1997), 1029–1033.
- [65] Van Santen, P.H. Jan, and Sperling G., *Elaborated Reichardt detectors*, *Journal of the Optical Society of America A* **2** (1985), 300–320.
- [66] R. Sarpeshkar, W. Bair, and C. Koch, *An analog VLSI chip for local velocity estimation based on reichardt's motion algorithm*, *Advances in Neural Information Processing Systems 5* (San Mateo, CA) (S.J. Hanson, J. Cowan, and L. Giles, eds.), Morgan Kaufmann, 1993, pp. 781–788.
- [67] R. Sarpeshkar, J. Kramer, G. Indiveri, and C. Koch, *Analog VLSI architectures for motion processing: From fundamental limits to system applications*, *Proceedings of the IEEE* **84** (1996), 969–987.
- [68] Liu S.C., *Silicon retina with adaptive filtering properties*, *Analog Integrated Circuits and Signal Processing* **18** (1999), 1–12.
- [69] E.C. Sobel, *The locust's use of motion parallax to measure distance*, *J. Comp. Physiol A.* **167** (1959), 579–588.
- [70] P.J. Sobey, *Active navigation with a monocular robot*, *Biol. Cybern* **71** (1994), 433–440.
- [71] M.V. Srinivasan and M. Lehrer, *Temporal acuity of honeybee vision: behavioral studies using moving stimuli*.

- [72] M.V. Srinivasan, M. Lehrer, W.H. Kirchner, and S.W. Zhang, *Range perception through apparent image speed in freely-flying honeybees*.
- [73] M.V. Srinivasan and S.W. Zhang, *Evidence for two distinct movement-detecting mechanisms in insect vision*, **80** (1993), 38–41.
- [74] ———, *Visual control of honeybee flight*, Orientation and communication in arthropods (M Lehrer, ed.), 1997, pp. 95–113.
- [75] M.V. Srinivasan, S.W. Zhang, and N. Bidwell, *Visually mediated odometry in honeybees*.
- [76] A. Starzyk, J. and X. Fang, *CMOS current mode winner-take-all circuit with both excitatory and inhibitory feedback*, Electronic Letters **29** (1993), no. 10, 908–910.
- [77] H. Sun and B. J. Frost, *Computation of different optic variables of looming objects in pigeon nucleus rotundus neurons*, Nat Neurosci **1** (1998), 296–303.
- [78] J. Tanner and C. Mead, *An integrated analog optical motion sensor*, VLSI Signal Processing, II (New York, ed.), IEEE Press, 1986, pp. 59–76.
- [79] Y. Taur and T.H. Ning, *Fundamentals of modern VLSI devices*, ch. 3, Cambridge University Press, 1998.
- [80] C. Toumazou, F.J. Lidgley, D.G. Haigh, and P. Peregrinus, *Analogue IC design: the current-mode approach*, ch. 2, Peter Peregrinus Ltd., 1990.
- [81] P. Venier, A. Mortara, X. Arreguit, and E. Vittoz, *An integrated cortical layer for orientation enhancement*, IEEE Journal of Solid State Circuits **32** (1997), no. 2, 177–186.
- [82] G.K. Wallace, *Visual scanning in the desert locust Schistocerca Gregaria Forskal*, J. Exp. iol. **36** (1959), 512–525.
- [83] Y. Wang and B. J. Frost, *Time to collision is signalled by neurons in the nucleus rotundus of pigeons*, Nature **356** (1992), 236–238.
- [84] L. Watts, D.A. Kerns, R.F. Lyon, and C.A. Mead, *Improved implementation of the silicon cochlea*, IEEE Journal of Solid State Circuits **27** (1992), 692–700.
- [85] K. Weber, S. Venkatesh, and M.V. Srinivasan, *Insect inspired behaviours for autonomous control of mobile robots*, Oxford University Press, Oxford, 1997.

# Appendix A

## Chip Layouts

In this thesis, we described two different monolithic implementations. Firstly, the hysteretic winner-take-all based visual motion sensors were discussed in Section 3.1 and the circuit details were described in Section 3.1.2. The two pixel level layouts, corresponding to two versions of the algorithm, are shown in Figure A.1 and A.2. The layout of the fabricated chip is shown in Figure A.3. The non-linear differentiator based motion sensor was described in Section 3.2 and its circuit was discussed in Section 3.2.2. The pixel level layouts of the algorithm are shown in Figure A.4, A.5 and A.6 and the chip layout is shown in Figure A.7, containing all three different version of the algorithm.

Three intensity based multi-chip implementations are described in the third chapter. First, the circuit details of the fabricated sender chip were given in Section 4.2 and its pixel and chip layouts are shown in Figures A.8 and A.9, respectively. Second, in Section 4.3.1 we discussed the Hassenstein-Reichardt correlation model and the details of its implementation with a multi-chip system were given in Section 4.3.1.3. The layout of the pixel in the fabricated receiver chip is illustrated in Figure A.10, and the whole chip in Figure A.11. After that we described the Adelson-Bergen spatiotemporal energy model in Section 4.3.2 and explained the implementation details in 4.3.2.4. This sensor has two fabricated versions and their pixel and chip layouts are shown in Figures A.12, A.13, A.14, and A.15. We explained the Barlow-Levick algorithm in Section 4.3.3. The pixel and chip layouts of this algorithm are shown in Figures A.16 and A.17, respectively. Lastly, the pixel and chip layouts of the centering behavior implementation are shown in Figures A.18 and A.19, respectively.

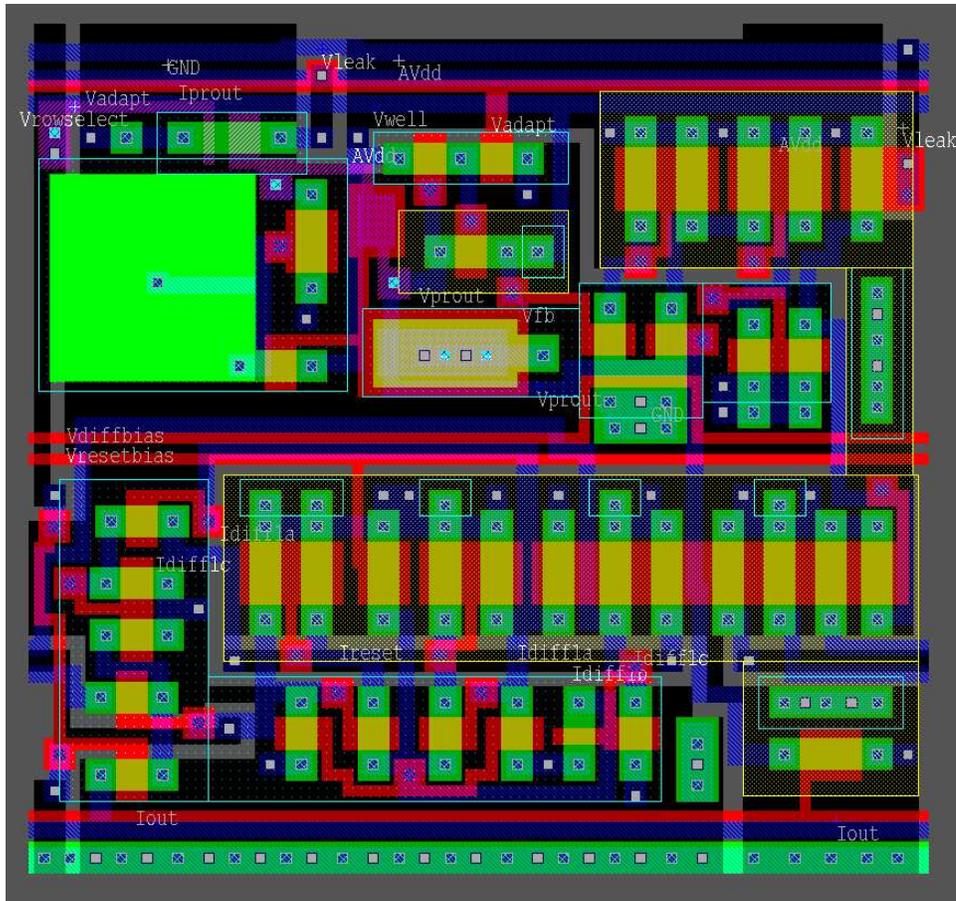


Figure A.1: The layout of the first version of the pixel level implementation of the hysteresis winner-take-all based motion sensor.



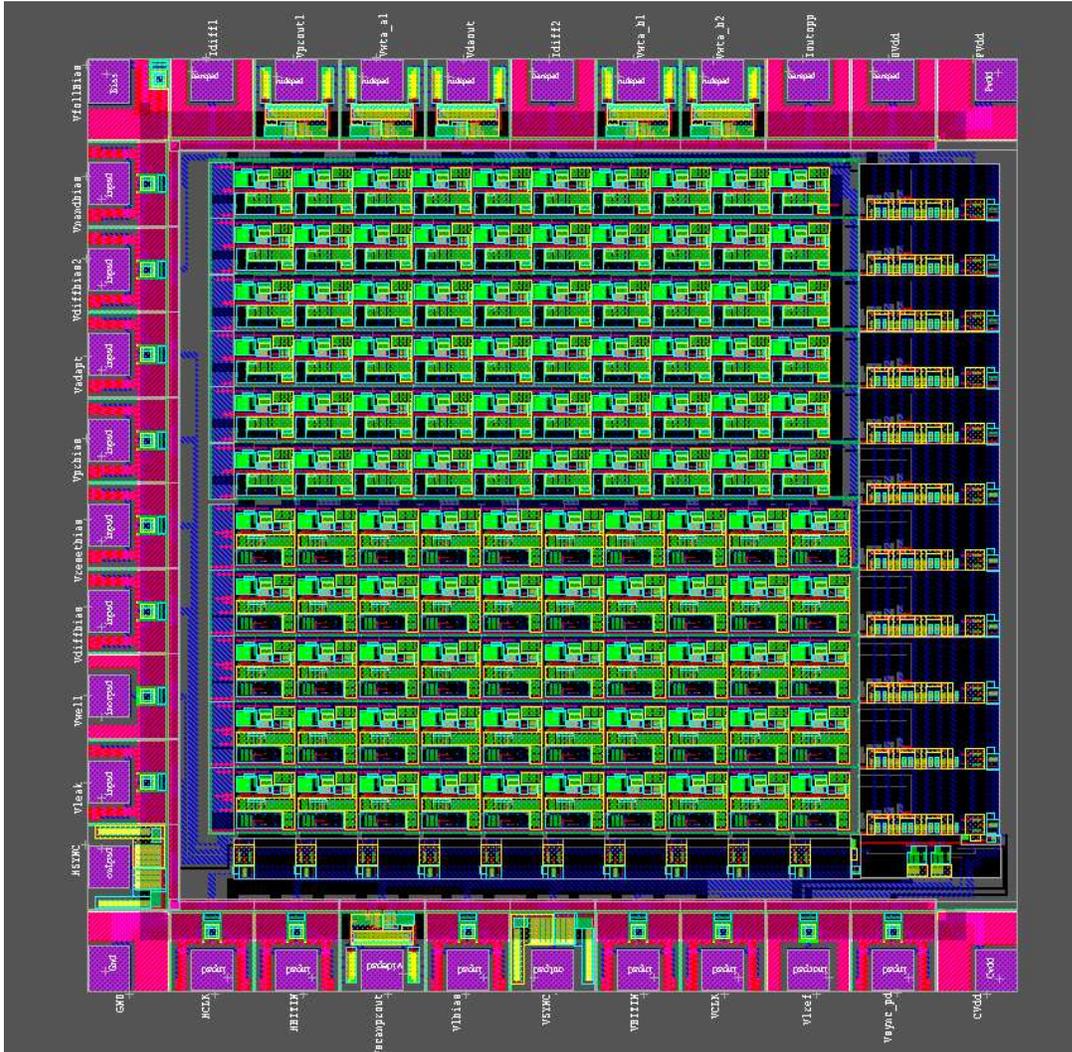


Figure A.3: The final chip layout of the hysteresis winner-take-all based motion sensor. This visual motion sensor is called WTA.

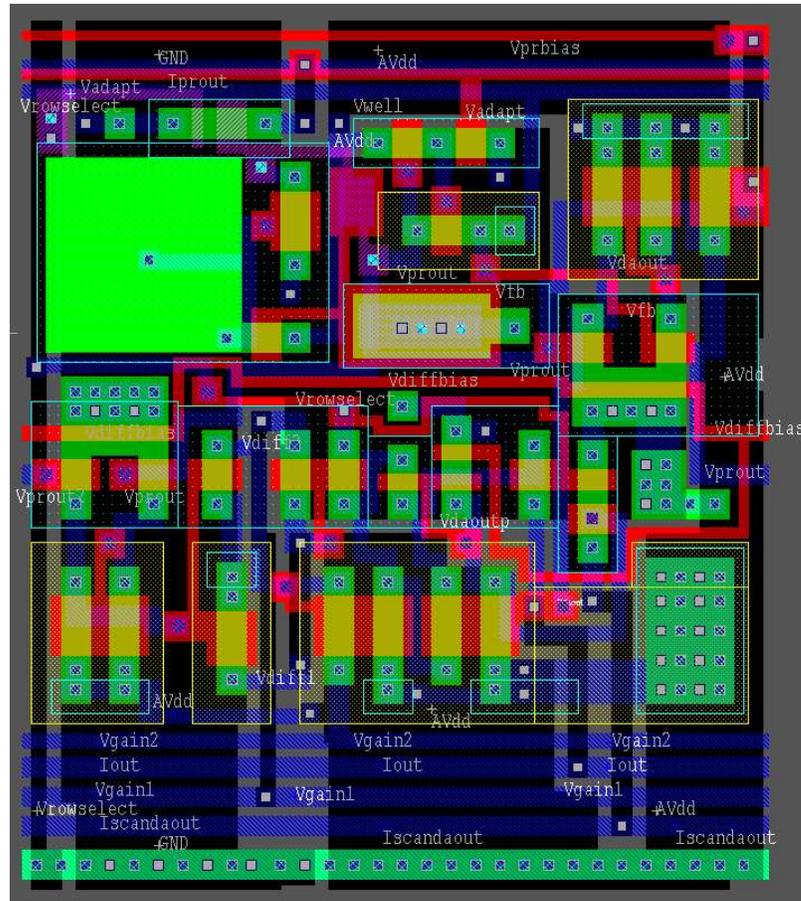


Figure A.4: The layout of the first version of the nonlinear differentiator based motion sensor.

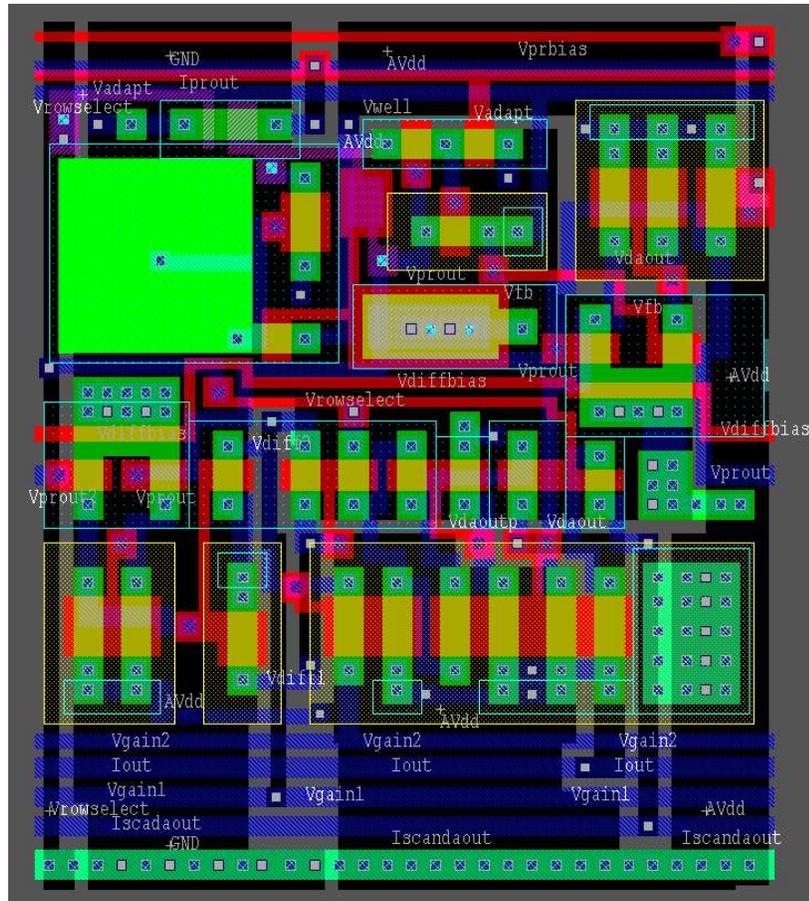


Figure A.5: The layout of the second version of the nonlinear differentiator based motion sensor.

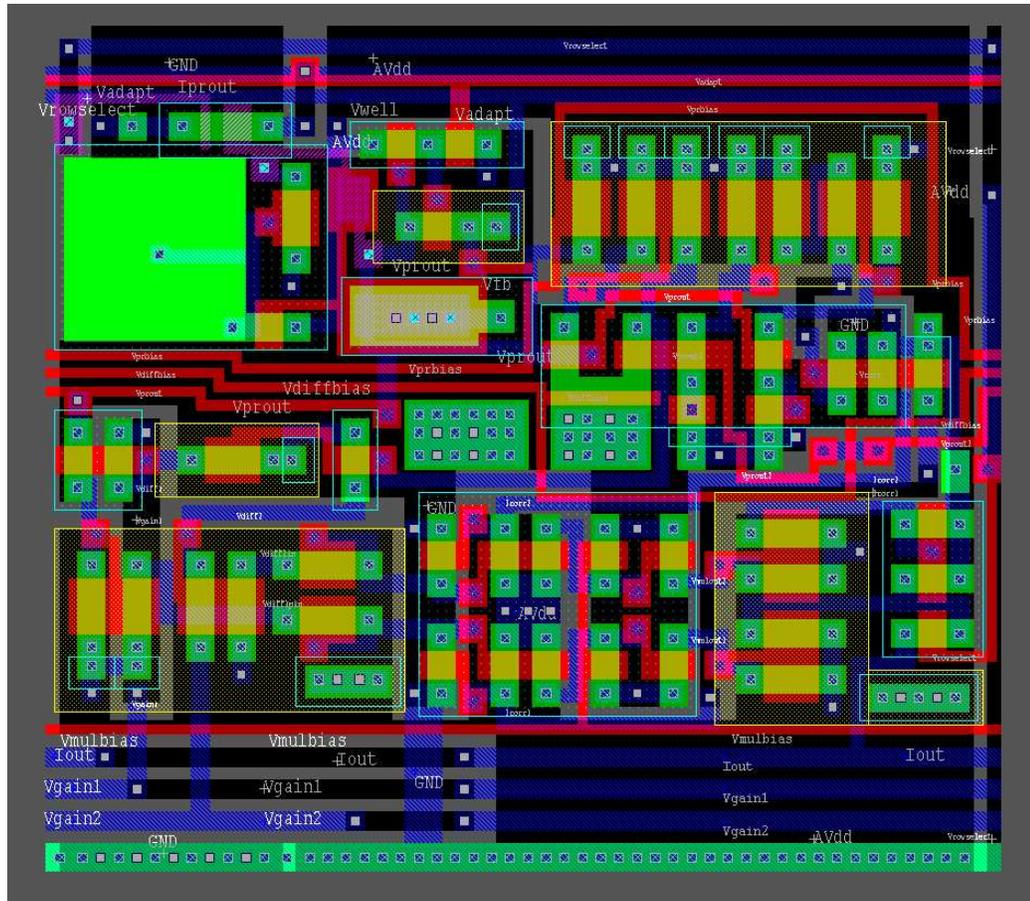


Figure A.6: The layout of the third version of the nonlinear differentiator based motion sensor.

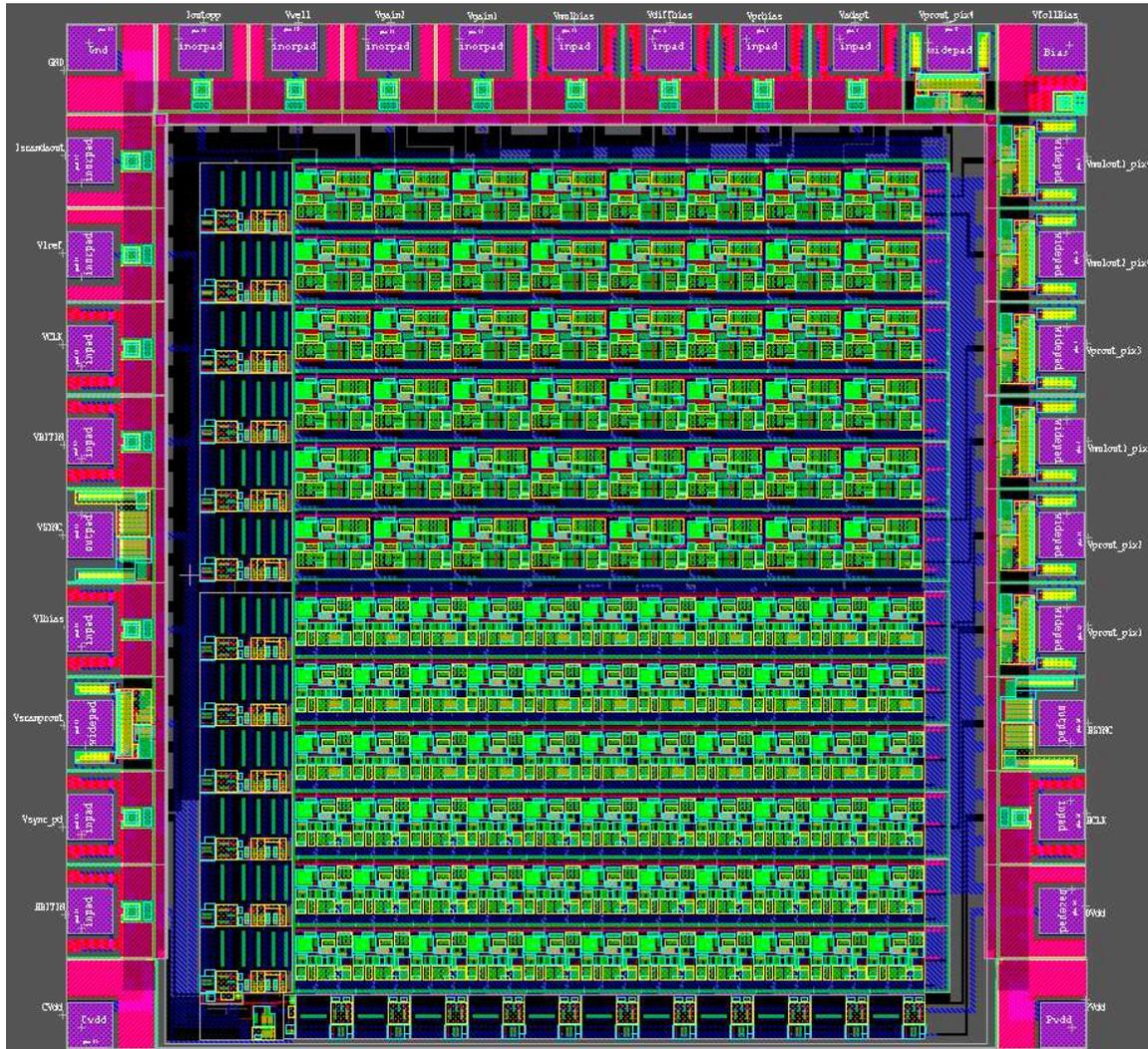


Figure A.7: The final chip layout of the nonlinear differentiator based motion sensor. This chip's name is NLD.

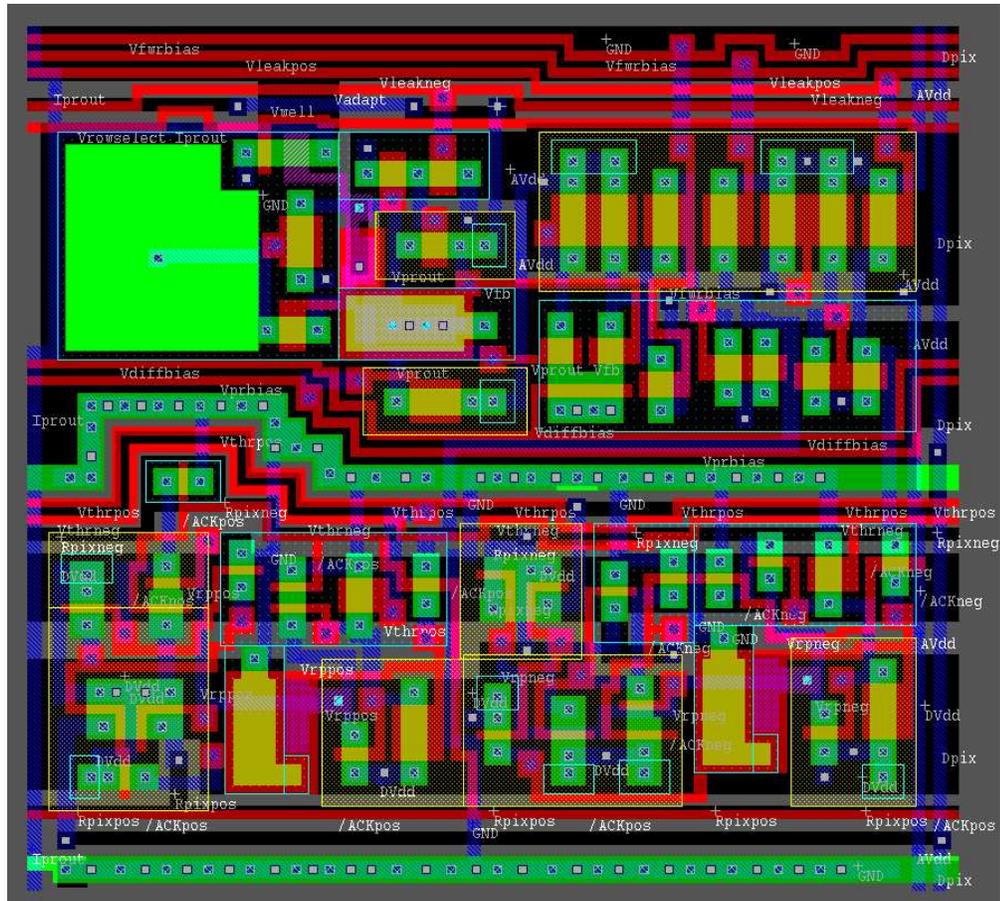


Figure A.8: The layout of sender pixel used for multi-chip implementations.

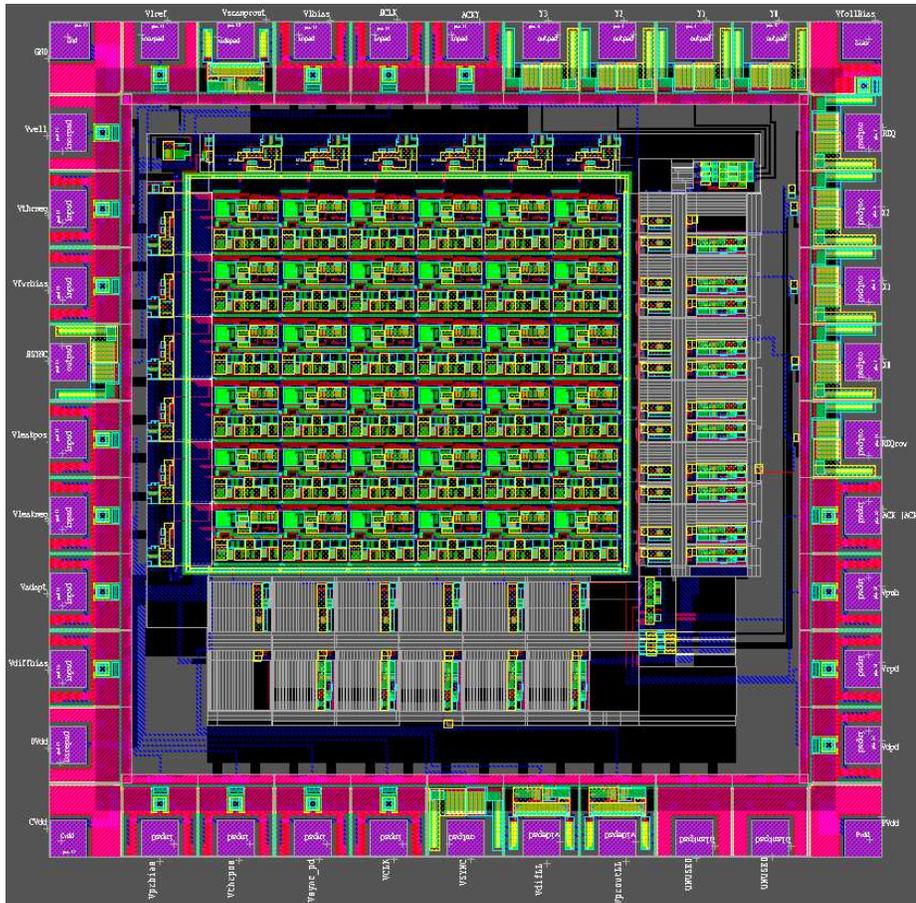


Figure A.9: The final layout of the sender chip. The name of this chip is ABSndr3.









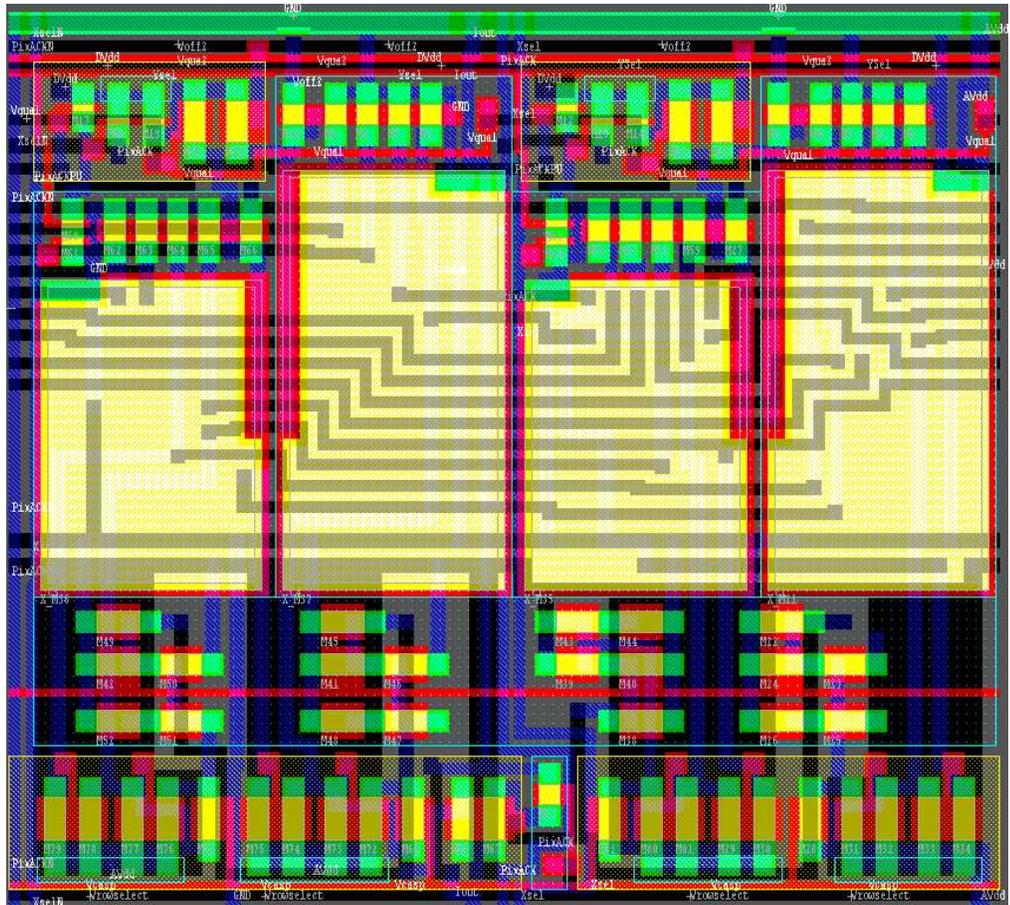


Figure A.14: The pixel layout of the second version of the Adelson-Bergen receiver chip.

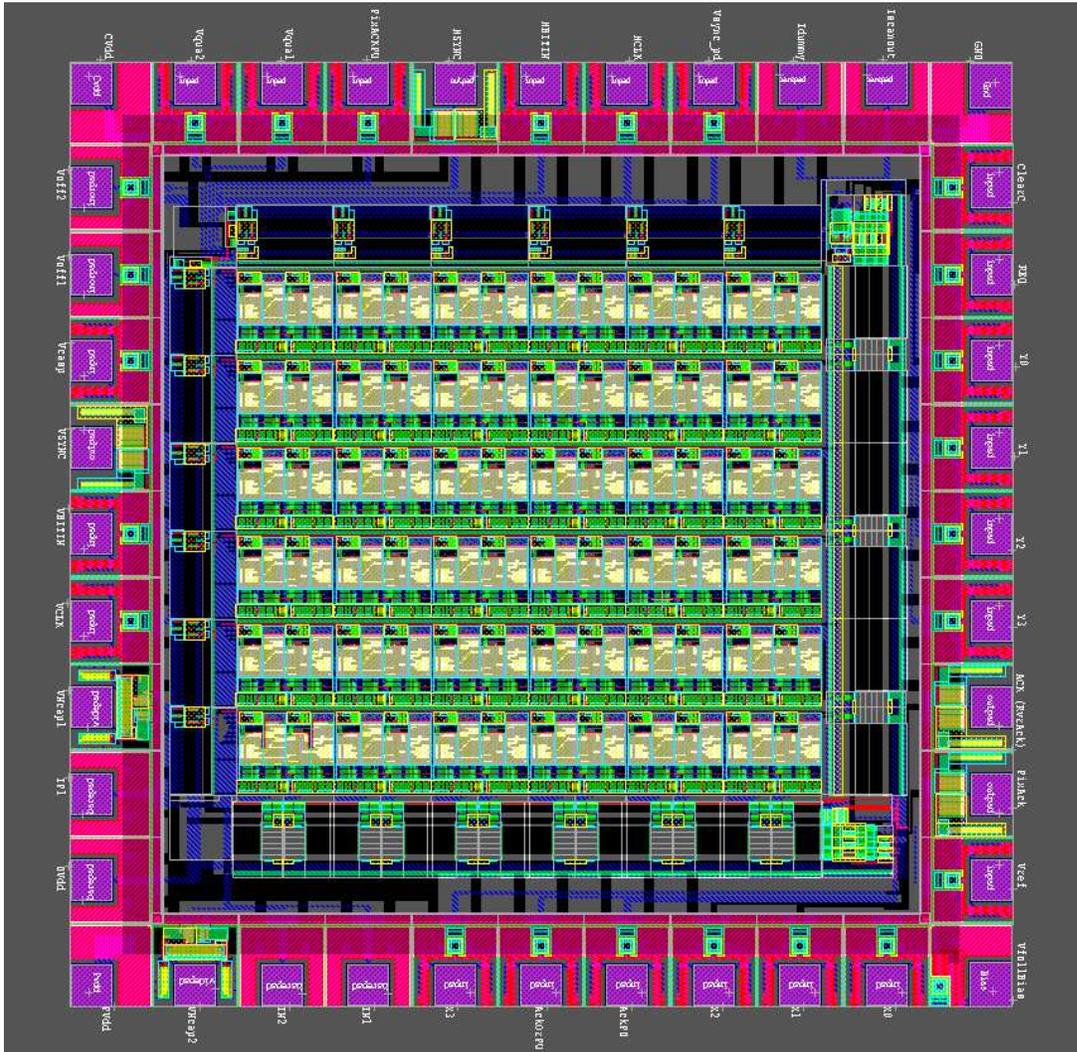


Figure A.15: The second version of the receiver chip layout that realizes the Adelson-Bergen algorithm. This chip is called ABmrcvr3a.



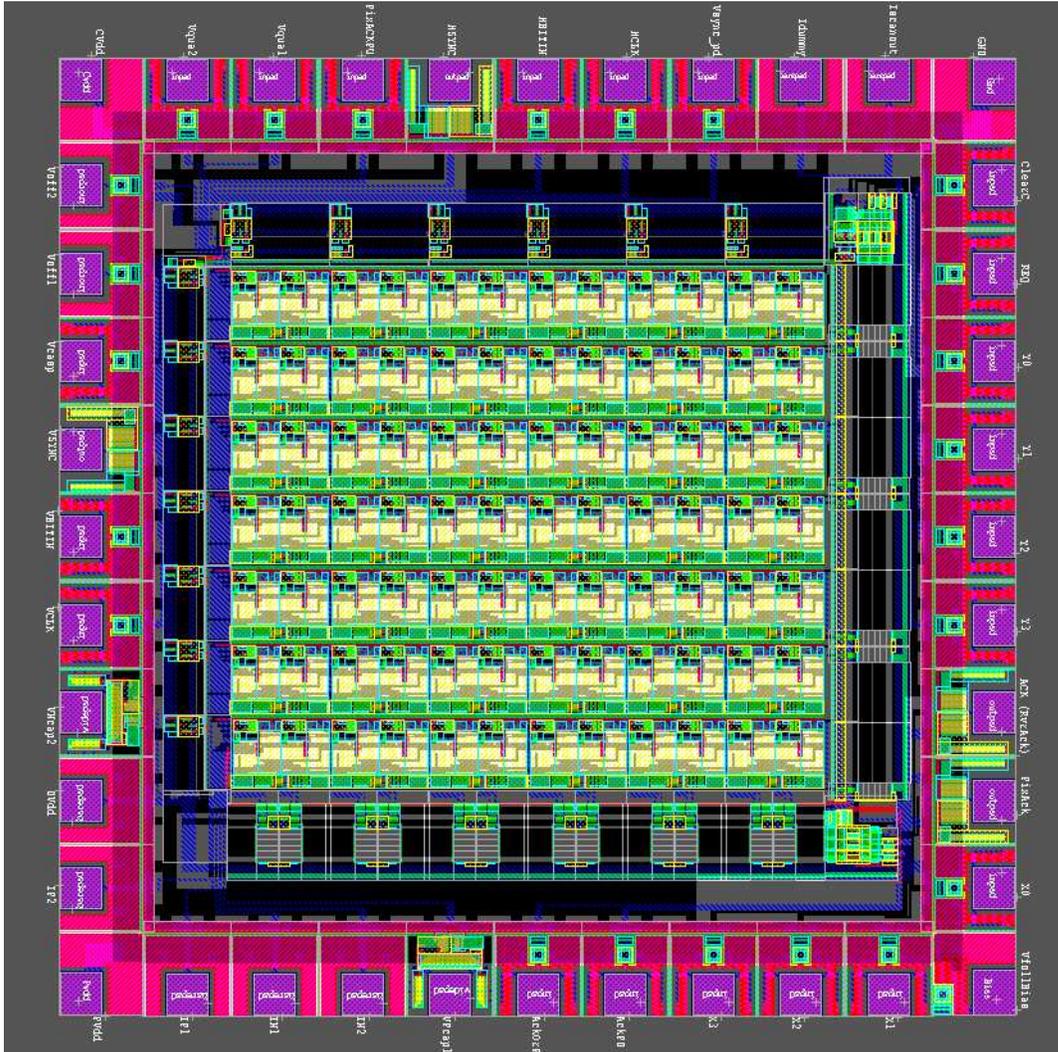


Figure A.17: The receiver chip layout that implements the Barlow-Levick model. The chip name is BL\_rcvr.

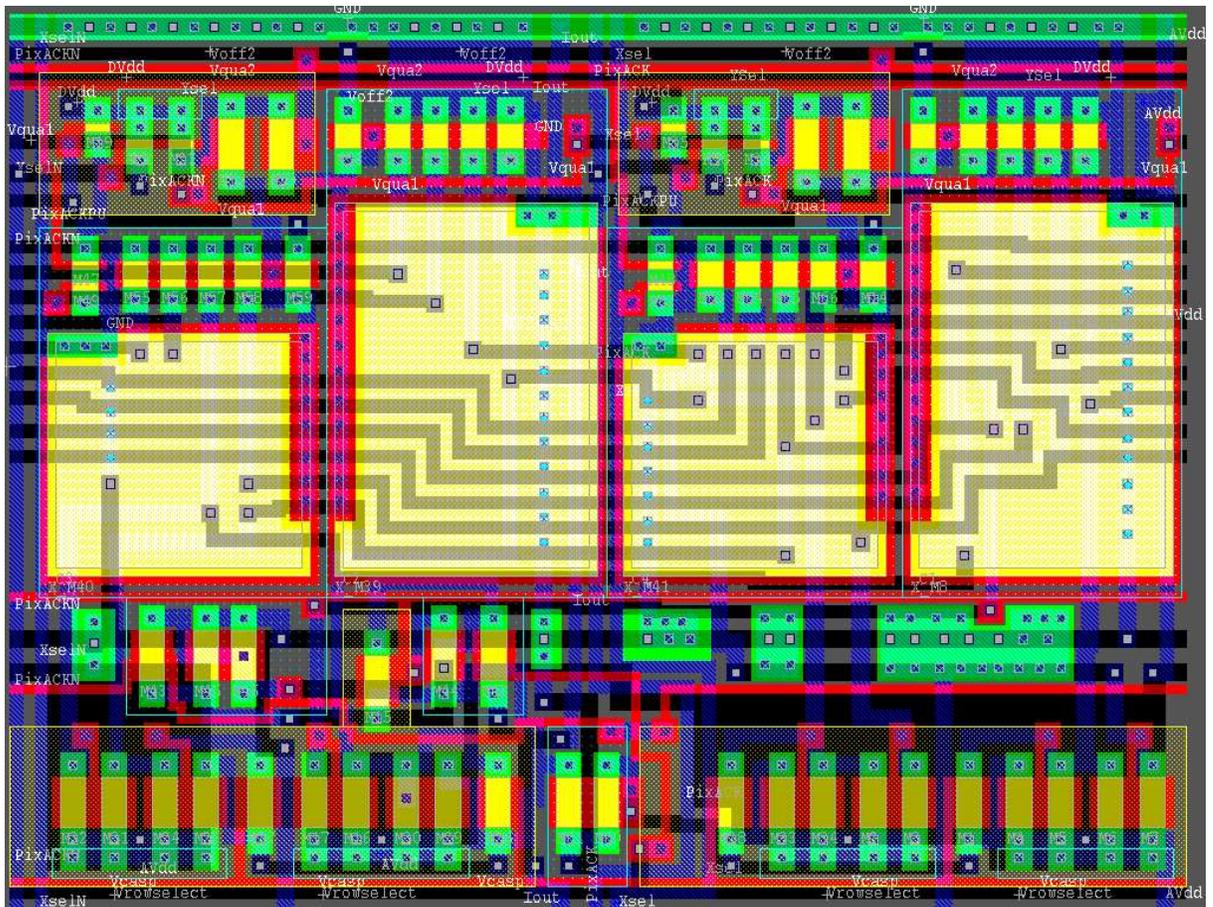


Figure A.18: The layout of the centering response receiver pixel.

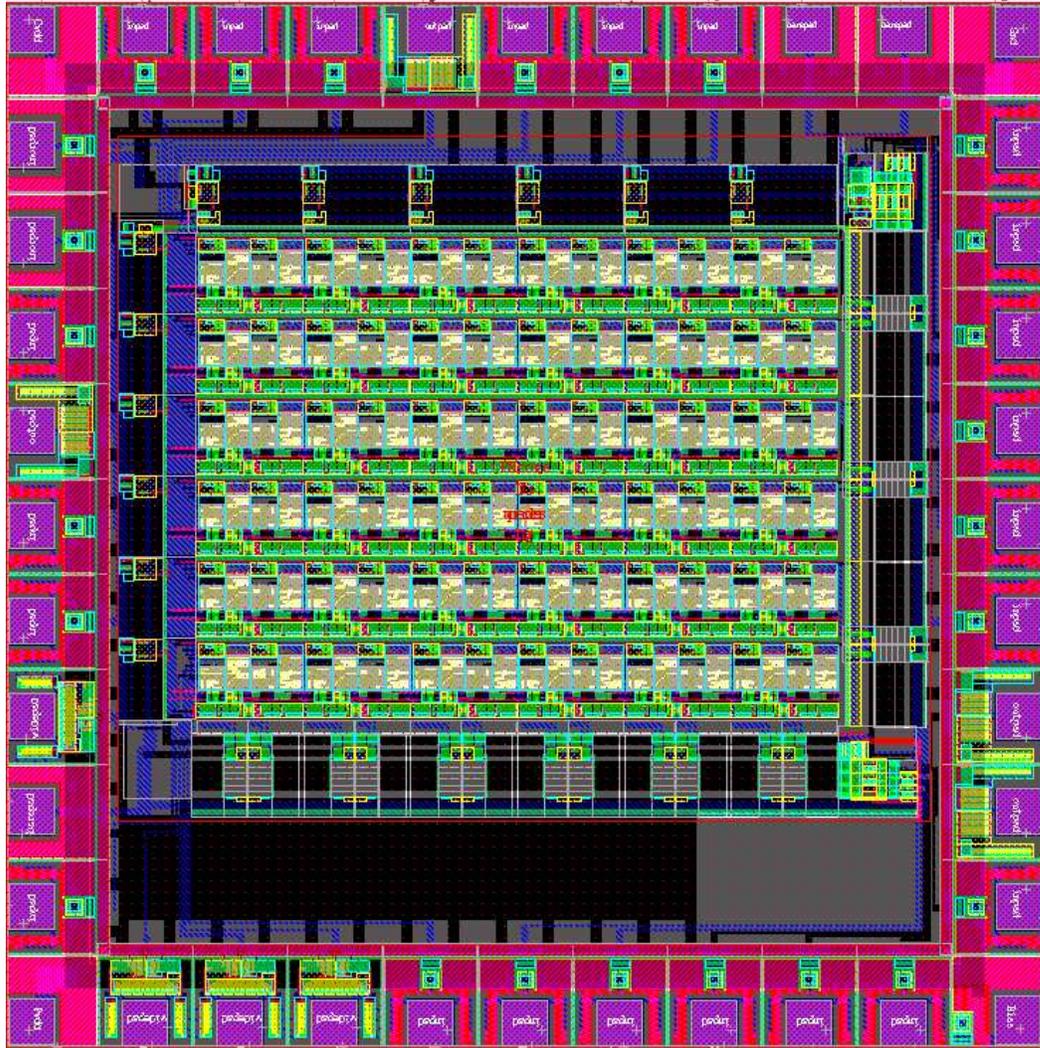


Figure A.19: The final layout implementation of the centering response receiver chip. This chip is called ObsavABrcvr.

## Appendix B

# Chip biases

In this part, we provide the biases of the chips that have been tested and characterized. First, we give a set of reliable biases that make robust multi-chip realizations possible. Second, the Adelson-Bergen receiver chip biases are given. In order to change the spatiotemporal tuning of this sensor  $V_{qua}$  and  $V_{off}$  values have to be adjusted properly. Next, we provide biases of the Hassenstein-Reichardt receiver chip. In contrast to Adelson-Bergen and Barlow-Levick, this sensor employs four-quadrant multipliers to obtain a nonlinearity that is required for motion computation. Similar to the Adelson-Bergen receiver chip, the spatiotemporal tuning can be changed by altering the  $V_{qua}$  and  $V_{off}$  biases. Lastly, we provide the Barlow-Levick receiver chip biases. This chip is also a spatiotemporally tuned motion sensor and it can be tuned for frequency by finding the right  $V_{qua}$  and  $V_{off}$  biases. Here we provided one such set of biases that helps the system yield a reliable motion output.

## B.1 Sender chip biases

Chip name: ABSndr3

- $V_{thrpos} = 1.112V$
- $V_{thrneg} = 1.088V$
- $V_{fwrbias} = 2.5V$
- $V_{leakpos} = 4.8V$
- $V_{leakneg} = 4.8V$
- $V_{adapt} = 0.126$
- $V_{prbias} = 3.51V$
- $V_{diffbias} = 0.354V$
- $V_{well} = 1.516V$
- $V_{lref} = 2V$
- $V_{follbias} = 0.788V$
- $V_{lbias} = 0.546V$
- $V_{pub} = 3.57V$
- $V_{rpd} = 5V$
- $V_{dpd} = 1.080V$
- $V_{syncpd} = 0.792V$

## B.2 Adelson-Bergen receiver chip biases

Chip name: ABmrcvr3

- $V_{off2} = 0.281V$
- $V_{off1} = 0.246V$
- $V_{syncpd} = 0.803V$
- $PIXACKPU = 3.81V$
- $V_{qua1} = 4.11V$
- $V_{qua2} = 4.17V$
- $V_{casp} = 3.34V$
- $ACKorPU = 3.99V$
- $ACKPd = 1.085V$
- $V_{follbias} = 0.786V$

### B.3 Hassenstein-Reichardt receiver chip biases

Chip name: Reichardt-rcvr2

- $V_{off2} = 0.218\text{V}$
- $V_{off1} = 0.219\text{V}$
- $V_{syncpd} = 0.790\text{V}$
- $PIXACKPU = 3.98\text{V}$
- $V_{qua1} = 4.13\text{V}$
- $V_{qua2} = 4.18\text{V}$
- $V_{refpos} = 4.29\text{V}$
- $V_{refneg} = 4.25$
- $ACKorPU = 3.99\text{V}$
- $ACKPd = 1.085\text{V}$
- $V_{follbias} = 0.787\text{V}$
- $V_{lref} = 1.916$
- $V_{bias} = 4.3$

## B.4 Barlow-Levick receiver chip biases

Chip name: BarlowLevick-rcvr

- $V_{off2} = 0.259\text{V}$
- $V_{off1} = 0.218\text{V}$
- $V_{syncpd} = 0.790\text{V}$
- $PIXACKPU = 3.72\text{V}$
- $V_{qua1} = 4.13\text{V}$
- $V_{qua2} = 4.18\text{V}$
- $V_{casp} = 3.5\text{V}$
- $ACKorPU = 3.98\text{V}$
- $ACKPd = 1.086\text{V}$
- $V_{follbias} = 0.787\text{V}$
- $V_{lref} = 2\text{V}$
- $V_{compref} = 2.5\text{V}$