Synthesis of Biological Vision Models Using Analog VLSI

Alireza Moini

Thesis submitted for the degree of
Doctor of Philosophy
in
Department of Electrical and Electronic Engineering
The University of Adelaide

Supervisors: **Dr. Abdesselam Bouzerdoum,**
Professor Kamran Eshraghian

1997

Contents

List of Figures

List of Tables

Abstract

Biological neural systems have been an inspiration for the implementation of artificial intelligent systems. The analog VLSI (AVLSI) design methodology tries to emulate the functionality of biological systems. This dissertation concerns the implementation of biologically oriented computer vision algorithms using silicon integrated circuit technologies. In addition to presenting the specific outcomes of the research, effort has been made to solidify the knowledge in the area into a systematic design methodology, and to identify and highlight some of the neglected drawbacks in the area of AVLSI, as future development of systems using AVLSI concepts will heavily rely on addressing these points.

Two specific early vision tasks, motion detection and contrast enhancement, which have been implemented during this research, are reviewed. VLSI design requirements for implementing biological vision are recognized. The capabilities of current VLSI technologies for implementing these systems, and circuit design methodologies suitable for this task are described.

A systematic view of all design levels, from the pixel level to the architectural level of vision chips are presented. Important issues in the design of analog VLSI (AVLSI) vision chips, including mismatch and digital noise, are highlighted and addressed.

Several vision chips have been designed, fabricated, and tested during the course of this research. Some of these chips are based on models of the motion detection mechanisms in the fly visual system, more specifically the "Template model". These motion detection chips are designed in a hybrid analog/digital design framework. One chip also explores shunting inhibition models of the early visual layers in the retina, and shares many of the characteristics of the natural counterparts. This chip is designed within the AVLSI design framework using subthreshold CMOS devices. The performance of new circuits for analog front end processing are evaluated using several test structure chips. Test results from these chips are used to derive conclusions regarding the suitability of the design techniques and circuits used in the implementation.

Statement of Originality

I hereby declare that this work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text.

I give consent to this copy of my thesis, when deposited in the University Library, being available for loan and photocopying.

Alireza Moini
20 December 1997

Acknowledgments

First of all I should thank and praise almighty Allah for his mercy upon me and guiding me through the world of unknowns.

My supervisors have been the greatest support for me during my postgraduate research. Dr. Abdesselam Bouzerdoum has been always there to answer my questions. His hardworking nature, which I could see present at all weekends, taught me one of the simplest and most important fundamentals of research; you cannot achieve anything by being normal, you should put sweat and blood into it. Professor Kamran Eshraghian gave me the confidence and the vision to see different perspectives of a new frontier in VLSI design.

My colleagues in the Vision Lab, Andre Yakovleff, Thong Nguyen, Richard Beare, and Andrew Blanksby are just exceptional *Aussie mates* always ready to be bugged by my questions. They all have a great share in this research and in the design of our vision chips.

Fellows in the VLSI lab, Andrew Beaumont-Smith, Said Al-Sarawi, Song Cui, Joan Liu, and Sam Appleton, have been helping me in many ways, by discussing circuits and layouts, helping me to learn new software, and offering me their friendship.

Richard and Andre went through the painstaking process of correcting the first draft of this thesis, and have a great share in making it readable for others.

Some of the new ideas implemented during this research were originated from my colleagues. The MNC operation was proposed by Richard. The CLM-OTA also came out of discussions with him.

My Iranian friends in the department, Abbas Zahedi Kouzani, Habib Mir Mohammad Hosseini, Morteza Biglari Abhari, Nozar Tabrizi, Mohammad Tahmasbi, Mohammad Feyzi, Mohammad Ali Tinati, Nasser Asgari, and Ali Sayyah Afkari, have also been of a great encouragement and support to me.

I must express my gratitude to my mother and father, and my brothers and sisters who have been my constant support, in every stage of my life.

Finally the love, passion, and patience that my wife, Leila, offered me during the writing of this thesis, goes beyond words.

Publications[1]

1. **A. Moini**, A. Bouzerdoum, K. Eshraghian, A. Yakovleff, X.T. Nguyen, A. Blanksby, R. Beare, D. Abbott, and R.E. Bogner, "An insect vision-based motion detection chip", *IEEE J. Solid State Circuits*, Vol. 32, No. 2, pp. 279–284, February 1997.

2. A. Yakovleff and **A. Moini**, "Motion perception using analog VLSI," To be published in *Journal of Analog Integrated Circuits and Signal Processing,* Kluwer Academic Publishers, 1998.

3. **A. Moini**, "Vision chips or seeing silicon: third revision", Tech. Rep., The Centre for High Performance Integrated Technologies and Systems, The University of Adelaide, Mar. 1997.

4. **A. Moini**, "Vision chips or seeing silicon: second revision", Tech. Rep., Centre for GaAs VLSI Technology, The University of Adelaide, Jan. 1996.

5. **A. Moini**, "Vision chips or seeing silicon", Tech. Rep., Centre for GaAs VLSI Technology, The University of Adelaide, Jan. 1995.

6. **A. Moini** and A. Bouzerdoum, "A biologically motivated imager and motion detector with pixel level image processing", *Australian Microelectronics Conference*, Melbourne, Australia, pp. 180–185, 1997.

7. **A. Moini**, A. Bouzerdoum and K. Eshraghian, "A current mode implementation of shunting inhibition", *ISCAS'97*, June 9–12, Hong Kong, pp. 557–560, 1997.

8. **A. Moini**, A. Bouzerdoum, A. Yakovleff, and K. Eshraghian, "A two-dimensional motion detector based on the insect vision", *SPIE, Advanced Focal Plane Arrays and Electronic Cameras, Berlin*, pp. 146–157, 10-11th Oct. 1996.

9. **A. Moini**, A. Blanksby, R. Beare, A. Bouzerdoum, and K. Eshraghian, "Multiplicative noise cancellation (MNC) in analog VLSI vision sensors", *Electronics Technology Directions for the Year 2000 (ETD2000), Adelaide, Australia*, pp. 253–257, 1995.

[1]The "Centre for GaAs VLSI Technology" changed name to "The Centre for High Performance Integrated Technologies and Systems" in 1996.

10. **A. Moini**, K. Eshraghian, and A. Bouzerdoum, "The impact of VLSI technologies on neural networks", *IEEE Int. Conf. Neural Networks*, pp. 158–163, 1995.

11. **A. Moini**, A. Bouzerdoum, K. Eshraghian, A. Yakovleff, and X.T. Nguyen, "The architecture of an insect vision based VLSI motion detection chip", *Australian Microelectronics Conference*, Adelaide, Australia, pp. 68–73, 1995.

12. **A. Moini**, A. Bouzerdoum, K. Eshraghian, X.T. Nguyen, and A. Yakovleff, "The VLSI implementation of an insect vision based motion detection chip", *OFOGH, The Journal of Computer Science and Engineering,* vol. 2, no. 2, pp. 30–38, Summer 1995.

13. **A. Moini**, A., Bouzerdoum, A., Yakovleff, D. Abbott, O. Kim, K. Eshraghian, and R.E. Bogner, "An analog implementation of early visual processing in insects", *Proc. 1993 Int. Symposium on VLSI Technology, Systems, and Applications*, pp. 283–287, May 1993.

14. **A. Moini** and K. Eshraghian, "Packaging for very high speed VLSI", Tech. Rep., Centre for GaAs VLSI Technology, The University of Adelaide, May 1993.

15. **A. Moini** and A. Yakovleff, "Testing the Bugeye I chip", Tech. Rep., Centre for GaAs VLSI Technology, The University of Adelaide, July 1993.

16. **A. Moini**, "Bug-Eye II the second generation of a motion detection chip: technical & programming information", Tech. Rep., Centre for GaAs VLSI Technology, The University of Adelaide, Mar. 1995.

17. **A. Moini** and K. Eshraghian "I/O and bus structures in GaAs design", Tech. Rep., Centre for GaAs VLSI Technology, The University of Adelaide, Mar. 1993.

18. **A. Moini** and K. Eshraghian "On the scaling of VLSI devices and circuits", Tech. Rep., Centre for GaAs VLSI Technology, The University of Adelaide, May 1992.

19. **A. Moini**, "Statistical analysis of the accuracy of current-mirror structures", Tech. Rep., Centre for GaAs VLSI Technology, The University of Adelaide, Mar. 1995.

20. **Alireza Moini**, "Design of a VLSI motion detector based upon the insect visual system", Master's thesis, The University of Adelaide, 1994.

21. D. Abbott, **A. Moini**, A. Yakovleff, X.T. Nguyen, A. Blanksby, G. Kim, A. Bouzerdoum, and R.E. Bogner, "A new VLSI smart sensor for collision avoidance inspired by insect vision", *SPIE Proc. Intelligent Vehicle Highway Systems*, vol. 2344, pp. 105–115, 1994.

22. D. Abbott, A. Yakovleff, **A. Moini**, X.T. Nguyen, A. Blanksby, R. Beare, A. Beaumont-Smith, G. Kim, A. Bouzerdoum, R.E. Bogner, and K. Eshraghian, "Biologically inspired

obstacle avoidance – a technology independent paradigm", *SPIE Proc. Mobile Robots X*, vol. 2591, pp. 2–12, 1995.

23. A. Bouzerdoum, R.E. Bogner, K. Eshraghian, **A. Moini**, A. Yakovleff, and X.T. Nguyen, "A VLSI front-end visual processor based on insect vision", *Sensory Stratagems: From Living Eyes to Seeing Machines*, Canberra, Australia, Feb. 1993.

24. A. Bouzerdoum, **A. Moini**, A. Yakovleff, X.T. Nguyen, R.E. Bogner, and K. Eshraghian, "A smart visual micro-sensor", *Proc. Int. IEEE Conf. on Systems, Man, and Cybernetics*, pp. 276–279, Oct. 1994.

25. X.T. Nguyen, A. Bouzerdoum, R.E. Bogner, K. Eshraghian, D. Abbott, and **A. Moini**, "The stair-step algorithm for velocity estimation", *First Australian and New Zealand Conference on Intelligent Information Systems*, pp. 412–416, Dec. 1993.

26. X.T. Nguyen, A. Yakovleff, **A. Moini**, K. Eshraghian, A. Bouzerdoum, and R.E. Bogner, "VLSI architecture of a low computation load processor for a visual system", *European Design & Test Conference*, pp. 85–89, 1994.

27. X.T. Nguyen, K. Eshraghian, **A. Moini**, A. Bouzerdoum, A. Yakovleff, D. Abbott, and R.E. Bogner, "An implementation of a smart visual micro-sensor based upon insect vision", *12th Australian Microelectronics Conference*, pp. 129–134, Oct. 1993.

28. X.T. Nguyen, A. Bouzerdoum, A. Yakovleff, **A. Moini**, R.E. Bogner, and K. Eshraghian, "A VLSI robotic micro-sensor: range-finder from self motion", *Proc. Int. Conf. on Machine Vision in Practice, Toowoomba, Australia*, pp. 78–83, Sept. 1994.

29. A. Yakovleff and **A. Moini**, "From seeing to perceiving: An overview of the Bugeye project", *SPIE, Advanced Focal Plane Arrays and Electronic Cameras, Berlin*, pp. –, 10-11th Oct. 1996.

30. A. Yakovleff, **A. Moini**, A. Bouzerdoum, X.T. Nguyen, R.E. Bogner, and K. Eshraghian, "A micro-sensor based on insect vision", *Workshop on Computer Architecture for Machine Perception*, pp. 137–146, Dec. 1993.

31. A. Yakovleff, X.T. Nguyen, A. Bouzerdoum, **A. Moini**, R.E. Bogner, and K. Eshraghian, "Dual-purpose interpretation of sensory information", *Proc. IEEE Int. Conf. Robotics & Automation*, pp. 1635–1640, May 1994.

Part I

Preface & Background

Chapter 1

Introduction

This chapter presents a set of introductory material, which in addition to providing a general view on the topic, highlights the importance of research in this area. It also presents a short history of the design of smart vision sensors, and points out some of the fundamental issues in the design of such sensors. The research conducted by the "Bugeye team" will be described, and its progress and research direction from its inception will be explained. And finally, the organization of the thesis is presented.

1.1 A General Overview

Machine vision is one of the main branches of artificial intelligence. The richness of information present in images makes them the first choice as an input to an artificial system which tries to interact with its environment. A large proportion of the brain of many advanced species is dedicated to visual information processing, which illustrates the importance of visual information in biological systems. Biological visual systems have evolved over millions of years, and each specie has developed a specialized visual system tailored for the essential tasks of survival, such as catching a prey, or escaping a predator. Implementing electronic hardware for image processing, therefore, may benefit from the underlying fundamental aspects of biological vision, though in no respect should this be regarded as a solid framework for electronic vision systems.

Traditionally, computer vision algorithms are performed on images captured by conventional cameras, and processing is accomplished by means of general purpose digital computers. More advanced systems utilize dedicated hardware to speed up the processing stage. However, in most systems the image capture and processing modules are viewed as two separate parts of the system, and often there is no interaction or relationship between these two stages.

Biological vision systems, on the other hand, have dedicated image capture layers (*retina*) in close interaction with subsequent stages in the neuro-optical pathways. Vision chips, which try to mimic this interaction by integrating both stages on the same chip, offer several advantages in comparison to the traditional camera-processor approach, including:

- **Speed and parallelism:** The processing speed achievable using vision chips exceeds that of the camera-processor combination. The main reason is the increased information transfer bandwidth between the imager and the processing elements which occurs in parallel, and also the reduction of the information to be transferred.

- **Large dynamic range:** Vision chips can provide adaptation to local light intensity, which increases the dynamic range. Also, by utilizing proper photodetector elements and photocircuits, the dynamic range can be increased even further.

- **Size:** By using a single chip implementation of image processing algorithms, compact systems can be realized. The only parts of the system that may not be scalable are the mechanical parts, such as the optical interface.

- **Power dissipation:** Vision chips often use analog circuits which operate in the subthreshold region, dissipating power in the order of several hundred nano-watts per circuit. There is also very little energy spent for transferring information from one level of processing to another.

- **System integration:** Vision chips may comprise most modules required for building a vision system. From a system design perspective this is a great advantage over the camera-processor option.

Although replicating biological vision on VLSI chips is an attractive solution to the gap between image capture and analysis, there are several limitations:

- The bulk of our understanding about most biological vision systems is obtained from controlled experiments on laboratory animals. The experiments are also mainly conducted on the early visual layers, as they are regularly organized, and are more accessible than deeper layers.

  Experiments on invertebrate subjects are more common than those on vertebrates, as they pose fewer "animal research ethics" problems, and do not require complicated neurosurgery to access the neuro-optical layers. In human subjects, almost all experiments on the performance of the visual system have been based on psychophysical techniques. In these experiments, special stimuli are carefully selected, in order to isolate different postulated mechanisms, and the subject is tested against these stimuli.

- The models devised and proposed to describe the behavior of biological vision have only recently started to solidify into rigorous mathematical formulations which can describe most aspects of biological vision. Still many biological models are empirical and experimental. Computational models for image processing, on the other hand, are based on mathematical foundations. They take into account a range of assumptions, or constraints, about the nature and contents of the image and noise. Computational models are often too complex for implementation even on powerful workstation computers to run in real-time.

- Implementation issues are the dominant barriers in developing VLSI hardware for visual processing. Algorithm selection, hardware design methodology, and system level integration are some of the high-level issues in hardware implementation of vision models. Low-level implementation issues, which are mainly related to circuit design, still require fundamental research to address problems such as dynamic range, mismatch, and digital and intrinsic noise. Another issue which directly affects the low-level aspects in the design of vision chips is the availability of different components, such as photodetectors and circuits for spatial and temporal processing of images.

- There is still a large gap between the achievable densities in VLSI and biological vision systems. This is mainly due to interconnection density limitations. In biological retinas information flows through a three-dimensional structure, while in silicon retinas the processed image should be transferred to other processing layers through a two-dimensional boundary.

This brief and general introduction was intended to highlight major issues that one should consider when designing a VLSI chip for vision processing. This area remains challenging as most of the issues mentioned have not been properly addressed or analyzed. Therefore, there remain vast and important areas of research on designing vision chips to be explored.

1.2 Vision Chips in the Past and Present

In this section an overview of the history and the current status of vision chip design is presented. As we will see, originally a mere implementation of a particular model in silicon was the major goal. Later research was directed more toward circuit level issues, along with system level issues, such as data transfer protocols between multiple chips.

Using networks of analog circuits for image processing can be regarded as the first step in the history of vision chips. Runge et al. in 1966 implemented an electronic retina using discrete components, mainly resistors and transistors, to model the pigeon retina [Runge et al. 68]. Fukushima et al. in 1970 designed another discrete analog network consisting of 700 photoreceptors, which was modeled after the retina [Fukushima et al. 70]. These implementations were aimed at simulating the processes that occur in the retina.

The accessibility of VLSI technologies in the late 70's initiated attempts to replicate neural circuitry on VLSI chips (see for example [Newcomb 81, Vidal 83]). Despite the rapid increase in analog VLSI and neural network research during the late 80's, only a few chips incorporated photodetectors, and could therefore be regarded as vision chips [Lyon 81, Forchheimer and Odmark 83, Tanner and Mead 84]. Most implementations were aimed at pure simulation of the neural layers in the retina using silicon hardware, rather than providing useful devices which could be utilized in real systems. This same trend could be seen in the "analog VLSI neural network" area.

The peak of research activity in the analog neural network area occurred in the early 90's, with a slight decrease during the last two years (95 and 96). Only very few analog neural network chips were introduced to the market, and had little success during this period. This clearly demonstrates that although the "analog VLSI neural network" framework may serve well for fundamental research on designing analog neural network chips, it cannot provide a firm basis for designing devices for use in real systems, as long as the issues highlighted in Chapter 6 are not addressed and answered.

1.2.1 Analog VLSI Design Framework

The most popular framework within which researchers have been implementing analog VLSI (AVLSI) chips, including vision chips, is the analog VLSI design framework, and is based on two main points (see for example [Mead 89]).

1. The cell complexity and accuracy (in the analog sense) of natural neural networks are usually low. They also possess a massively interconnected structure.

2. Analog VLSI implementation of neural networks can yield interconnected networks of simple functional units with a low precision.

From these points it was concluded that AVLSI circuits are able to simulate or replicate neural functionality. CMOS (Complementary Metal-Oxide-Semiconductor) devices operating in the subthreshold region were then recognized as one of the prime choices in designing circuits for this purpose, as they can provide a rich set of circuits with considerably less area and power consumption than digital or conventional analog implementations. At first glance this may seem to be a fair assumption, because of the basic similarities between the nature of some of the models and AVLSI. Unfortunately, these similarities cannot provide firm ground for fundamental research, as several issues are not addressed.

Firstly, the AVLSI framework is based on the similarities in the limitations of electronic hardware and some *postulated* weaknesses of biological neural systems, in particular low accuracy and interconnection limits. This tends not to be completely true. The interconnections in biological systems are three-dimensional, while current VLSI technologies are limited to planar structures. Also, while it may seem that precision in biological neural systems is relatively low, the overall performance is much higher than what can be achieved by the most reliable analog or digital technologies. The limits of operation of biological systems are often governed by fundamental *physical and mathematical* limits, not by *technological* ones.

Secondly, the AVLSI framework does not provide any criteria for assessing the performance of a limited-size neural network which uses neurons with limited precision and functionality. Thirdly, this framework does not answer the issues related to interfacing a particular neural network layer to its subsequent processing stages. For example, " Can the next stage use the noisy signals, which are contaminated by the circuit noise and mismatch?". The sequence of neural processing in natural neural networks has developed with the limitations of each stage being accounted for. Analog neural networks, if used in a real system, often require analog-to-digital and digital-to-analog converters at the interfacing points with digital processors. Processing the digitized output of a chip and trying to compensate for the added mismatch and circuit noise could be more difficult than dealing with raw but clean input data in the first place. Although an AVLSI system may ultimately incorporate all necessary stages from sensory systems to processing and motor control, current AVLSI systems do still have a long way from reaching this goal.

Vision chips suffer from the same problems as analog neural networks because they share the same structure and principles for processing the information. The only exception is that the input is provided by photoreceptors. The additional limitations of photoreceptors and photo-

circuits[1], have made the design of reliable vision chips even less successful than analog neural network chips.

1.3 History of Our Research

The original concept of implementing an insect vision based motion sensor was developed by Professor Adrian Horridge from Australian National University, Professor Robert Bogner and Professor Kamran Eshraghian, both from Adelaide University, in 1991. In 1992 the insect vision team (later named the Bugeye team) started working on different aspects of the project. The *Template model*, which represents a simplified model of the motion detection mechanisms found in insects, was used as the basis for the first VLSI chip. Various functional modules necessary for the implementation were then identified and designed. The VLSI design was completed in late 1992 and the chip was fabricated and tested in 1993.

The first chip, dubbed "Bugeye-I", was intended as a proof-of-concept design, to demonstrate the feasibility of integrating photodetection and phototransduction, analog preprocessing, and digital processing all on one chip. The chip functioned successfully under laboratory conditions, although there were several problems related to the analog circuits that needed to be addressed.

With the commercial support of an industry partner, the project aimed at developing a motion sensor for blind-spot detection in automobiles. The next chip, named "Bugeye-II", was designed as a test-bed for several analog circuits, which were modified versions of the circuits used in Bugeye-I. Bugeye-II comprised only the analog core. Most of the analog circuits implemented in the chip were successfully tested. However, the analog temporal differentiator remained unreliable and sensitive to lighting and biasing conditions.

"Bugeye-III" was designed as a test chip for various differentiator circuits. It contained 17 different temporal contrast detection (TCD) circuits. Comprehensive tests identified the best TCD circuit which could function reliably under large variations in lighting and biasing conditions. As a result, "Bugeye-IV" chip, which contained an array of 4×64 TCD circuits, was designed and fabricated.

Finally "Bugeye V", which comprises all necessary modules for implementing the template model was designed and fabricated. It contains two arrays of photocircuits, whose outputs are passed to an on-chip ADC and then to a digital processing unit. The chip can be configured to provide different information, such as the raw image data, templates, local motion, or global motion, and its design is aimed at facilitating integration into real systems, such as robotics.

[1] I have adopted the term "photocircuit" because of its clear and sharp reference to a circuitry which processes the photocurrent or photovoltage. Other terms, such as "photoreceptor", have been interchangeably used both for single photodetectors and the circuitry used for processing photocurrents, and in a context full of these references can become confusing.

As a complement to the project, work was done on implementing the contrast enhancement and spatial adaptation capabilities of the retina. The "MNCSI" chip, contains several one-dimensional arrays of various models of the retina. Test results from this chip were used in implementing a two-dimensional smart sensor.

1.4 Organization of The Thesis

This thesis is organized in a top-down fashion, divided into four parts. Part I establishes the background, and contains basic material from the two different aspects of the project, i.e. computer vision and VLSI. In Chapter 2 models and algorithms for low-level motion detection and contrast enhancement are reviewed. Implementation requirements for these two vision tasks are discussed. In Chapter 3, VLSI technologies, analog design techniques and their capabilities for implementing computer vision algorithms are presented.

Part II is concerned with physical implementation issues at various levels. It addresses pixel-level physical design and tessellation strategies, and describes sensor-processor structures and system-level communication methods. It then presents the circuit level design of photocircuits, spatial, temporal, and spatio-temporal processing circuits. Also, some of the issues that severely affect the design of AVLSI circuits, such as mismatch and digital noise, are addressed in detail.

Part III describes three of the designed sensors and discusses the experimental results. The significance of Bugeye II, described in Chapter 8, is from a circuit design point of view. Bugeye V, described in Chapter 9, has an architectural significance. Finally, the shunting inhibition chip in Chapter 10 embodies novel AVLSI circuits for contrast enhancement.

In Part IV problems encountered during the research are presented, possible solutions are described, and future directions for continued work and the enhancement of this field of research are presented.

Finally, the derivations of some of the equations which appear in the main body of the thesis are relegated to in the appendices.

Chapter 2

Computer Vision for Analog VLSI

"Even the simplest brains of the simplest animals are awesome computational instruments. They do computations we do not know how to do, in ways we do not understand"

Carver Mead in IEEE Proceedings, 1990

In this chapter an overview of models for motion detection and contrast enhancement in early vision is presented. The reason for addressing only these two visual tasks is that, originally, the project aimed at designing a motion detection system based on the "Template model". Later we investigated the implementation of contrast enhancement models, in particular retinal shunting inhibition.

Although many other visual tasks can be studied for VLSI implementation, most of the issues are similar to those of the two specific tasks that are addressed here, i.e. motion detection and contrast enhancement, and many of the circuits and building blocks can be reused.

Introduction

Computer vision research is undoubtedly one of the most active areas in artificial intelligence. This branch of artificial intelligence can certainly act as the most useful source of information for an intelligent system which is intended to interact with its environment. A large body of the research in this area, however, is concentrated on developing algorithms, whether it be motion detection, object recognition, stereo vision, or any other type of visual information processing.

The biological approach toward computer vision tries to utilize some of the basic observations concerning biological vision systems in the development and implementation of vision algorithms and hardware. The two most recognized observations are the parallel nature of the visual *wetware*, and the relative simplicity of the processing elements in the visual pathways. These principles have been highlighted in many biological models for vision.

When developing algorithms, hardware related issues are often completely ignored. The driving force behind many algorithms is the reliability and flexibility, for operation in real environments, rather than physical implementation. As a result, many algorithms cannot run in real-time even on supercomputers.

While several vision algorithms could be found that have been tailored to digital implementations, there are almost no algorithms targeted at AVLSI. In fact, it has been assumed that the AVLSI hardware should match itself with the algorithm. At first glance, considering the basic similarities between the nature of some vision models (especially biological ones) and AVLSI, it may seem that this is a fair assumption. However, these similarities have not been fundamentally helpful in designing successful vision systems. For example, the architecture for many motion detection chips is based on the Hassenstein-Reichardt model [Hassenstein and Reichardt 56]. When considering the function of a single motion detection element, most implementations provide acceptable performance in comparison to the mathematical model. However, in a larger AVLSI system composed of many of these motion detection elements, the outputs of the elements become heavily dominated by circuit mismatches and other nonidealities in AVLSI.

In the following sections we focus on the models of two early vision tasks, i.e. *motion detection*, and *contrast enhancement*, which have been targeted for AVLSI implementation in this project. Computational and biological models are both briefly reviewed and presented, with particular emphasis on architectural and AVLSI implementation aspects.

2.1 Computational Motion Detection Algorithms

Motion cues are a very rich source of visual information. Segmenting objects, finding depth and other vital information, such as time-to-collision, can be obtained from motion information.

Distinction should be made between motion detection and velocity estimation. In motion detection only the existence and the direction of motion at each pixel are detected. In velocity estimation, the original velocity vectors of the object are estimated from the velocity vectors of the image, which is also known as *optical flow*. The object velocity field does not necessarily correspond to the optical flow field. Therefore, it is theoretically impossible to find the *exact* motion field. However, in some realistic cases close approximations of the motion field can be obtained.

There is a large number of computational models for motion detection. Detailed characteristics of each model can be found in the literature (see for example [Laplante and Stoyenko 96]). Here an overview of these models is presented.

Computational motion detection models can be categorized into three main groups: *intensity-based*, *feature-based*, and *correlation-based*. In intensity-based models local spatial and temporal gradients of the image are used to determine the optic flow and motion. In feature matching techniques features such as points, lines, and corners are found, and tracked over time. In correlation-based algorithms either the raw image or the processed image (for example after edge detection) is correlated with the previous frame to find the movement in the image. Due to the extremely high computational needs of correlation-based algorithms they are not as widely used as other algorithms.

The main problem of intensity-based techniques is known as the *aperture problem*, whereby the direction of motion of an object cannot be determined uniquely, when looked at from a small aperture. Feature matching models face the *correspondence* problem, where the features in one frame are to be matched with corresponding features in the previous frame. Feature matching techniques require image information over a wider area of the image, and are therefore more computationally expensive.

Intensity-based algorithms provide a better architecture for implementation, from an AVLSI point of view, because the information required for processing is often within a local neighborhood. In feature matching algorithms mechanisms for global searches and feature memory may be needed, depending on the type of features being detected. For example, if the features are *outlier pixels* or points, then an analog network with local interconnection might be able to locate them [Harris et al. 90]. Global correlation-based algorithms cannot be implemented in current analog networks, due to their high demand on memory and interconnection resources.

2.1.1 Intensity Based Motion Detection

The intensity gradient in time and space has been used to determine the optical flow, and motion field. In the method developed by Horn and Schunck [Horn and Schunck 81] it is assumed that the brightness is constant. Based on this assumption, the relationship between the velocity and intensity gradients has been derived as the brightness change constancy equation (BCCE)

$$\nabla \vec{I} \cdot \vec{V} = \frac{\partial I}{\partial t} \tag{2.1}$$

where $\nabla \vec{I}$ is the intensity gradient, $\vec{V}$ is the velocity vector, and $\partial I/\partial t$ is the temporal derivative of intensity. This equation applies only when the intensity changes are due to the translational motion and the image is smooth, except at a finite number of points. The smoothness constraint used by Horn & Schunck, is derived by minimizing the square of the magnitude of the gradient of the optical flow velocity

$$\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \tag{2.2}$$

where $\partial u/\partial x, \partial u/\partial y, \partial v/\partial x, \partial v/\partial y$ are the partial derivatives of the velocity vector. The smoothness error to be minimized is:

$$E_{sm} = \int\int \left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2\right] dx\, dy \tag{2.3}$$

There are a number of difficulties in applying gradient models to real image sequences. Firstly, the intensity constancy assumption is valid only under very limited conditions and is violated around the image boundaries and motion boundaries. Secondly, spatio-temporal derivatives produce noisy and unreliable data.

Others have tried to improve the estimation of the motion field by reducing the smoothing of the velocity field in areas close to the edges with high spatial gradients. For example, Hildreth [Hildreth 85] applies the smoothness constraint along image contours. The contours are zero-crossings determined by applying the LOG (Laplacian of Gaussian) operator. The smoothness error in this case is

$$E_{sm} = \oint \left|\frac{\partial u}{\partial s}\right|^2 ds \tag{2.4}$$

where s denotes the direction of the contour at a point on the contour.

Nagel has constrained the smoothing of the optical flow only in the direction perpendicular to the spatial gradient [Nagel 87]. He has further extended the constraint to spatio-temporal orientations [Nagel 90].

There are other methods which use affine flow-field models [Gupta and Kanal 97], wavelet transforms [Corghi et al. 97], multiresolution techniques [Memin and Perez 96,Ong and Spann 96],

block-matching [Musmann et al. 85, LeQuang and Zaccarin 97] (although block-matching techniques have been mainly used for motion estimation in video coding), and even feature-based methods [Buxton and Buxton 84] to find a reliable estimate for the optical flow field.

The advancement of optical-flow techniques has resulted in more complex algorithms which are often accompanied by more constraints and require more spatio-temporal information. These all make such techniques less tractable for analog VLSI implementations.

2.1.2 Feature Based Motion Detection

In real scenes a large proportion of the image may not contain *useful information*. Therefore, it would be of great advantage to focus on salient features, which provide the most useful knowledge about the scene. Feature-based techniques for motion detection thus reduce the amount of information by concentrating on the displacement of features of interest in a sequence of images. There are two major issues concerning this class of algorithms. Firstly, features must be reliably and accurately detected in each image frame. Secondly, the correspondence problem arises due to similarities between distinct features, which may confuse the process of tracking the displacement of one feature from an image to the next. Both feature detection and solving the correspondence problems are non-trivial tasks.

Some of the most commonly used features are: corners, edges, blobs, and templates. The first three features are purely spatial, but the fourth is a *spatio-temporal* feature, which we have used in our approach [Horridge 90].

Corners are two-dimensional features, which are mainly defined by points of *high curvature* in the image boundaries [Smith and Brady 97]. Many corner detection algorithms have low reliability and are sensitive to noise [Haralick 93]. *Interest points* have also been regarded as forming a superset of corners, although the definition of what really constitutes an interest point is somewhat open ended.

Edge elements and lines have not only been used as features for motion detection, but also for many other tasks, such as segmentation. Edge detection is one of the most developed tasks in computer vision, mainly due to its relative structural simplicity and clear definition. Many books on machine vision contain detailed analyses of various edge detection algorithms (see for example [Faugeras 93, Horn 86]). A classical mathematical method for edge detection has been the zero-crossings of the Laplacian of Gaussian (LoG) operator [Marr and Hildreth 80]. A large number of other methods exist, which present many improvements over the method based on LoG.

In the context of "feature for motion detection", edge elements and lines are more tractable than corners, as they have several parameters, such as line terminations, length, and direction, that can be matched between frames. In the case of corners, only the position and sometimes the type of corner (for example "L", "T", or "X" junctions) can be matched. Therefore, the type of

corner and its accurate location become critical parameters, especially in noisy environments.

There is a threshold level in all these models to validate the feature, below which a feature is not detected. Selecting the optimum threshold often requires *a priori* information about the noise and its characteristics, as well as the image contents. So it is not unexpected to observe that almost all models fail under some noise and signal (image) conditions.

2.1.3 Correlation Based Motion Detection

All previous models require some form of spatial and temporal derivatives of the intensity to compute motion. In intensity based methods these derivatives are directly involved in the processing. In feature based methods derivatives are involved indirectly, and are used for detecting features. In correlation based models, each frame is correlated with the previous frame. Movement in parts of the image can be detected by observing the location of the maximum correlation with respect to the previous frame. Due to their integrative rather than differentiative nature, correlation-based models produce more robust output, although the computational cost of the large number of correlation operations is often very high [Bohrer et al. 90].

In order to alleviate the computational cost of this approach, one may limit the image blocks to be correlated to a block of 1×1 (only one pixel) and only over a span of one pixel. In fact this extremely simplified approach is the essential building block in the correlation-based motion detection units proposed for the fly visual system [Reichardt and Egelhaaf 88].

2.2 Biological Motion Detection Models

Models proposed for biological motion detection fall mainly into the category of *local* models, where there is only interaction with the nearest neighbors. Although most models are based on time domain representation of motion, there also exist spatio-temporal energy models which measure the spatial and temporal energy of the visual field within a range of spatial and temporal frequencies [Adelson and Bergen 85].

The structure of most models are relatively simple, and are similar to or derived from the Reichardt detector, which in essence uses two branches of delay-and-compare elements to detect motion in two directions. The basic structure for an elementary motion detector (EMD) is shown in Figure 2.1-a. A bidirectional motion detector using two EMD cells can be constructed as shown in Figure 2.1-b. The delayed signal from one channel is compared to the signal from the other channel. The **M** element performs a nonlinear operation, so that directional selectivity is preserved. Two types of *veto* and *conjunctive* (or in more familiar words inhibitory and excitatory) elements can be used [Barlow and Lewick 65, Hassenstein and Reichardt 56]. Early studies tried to describe the function of the elementary motion detection units by using either inhibitory or excitatory interactions [Franceschini et al. 89, Schmid and Bulthoff 88]. However, Bouzerdoum suggested that both mechanisms could exist, and depending on the order and spatial spread of the stimuli used in the experiments, only one mechanism may be excited or be dominant [Bouzerdoum and Pinter 92].

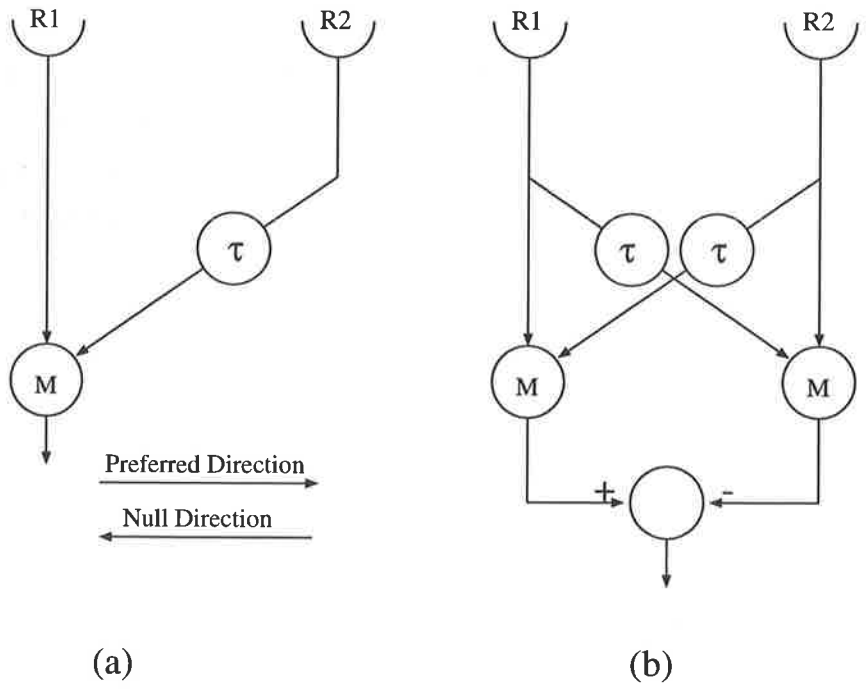


Figure 2.1: a) The elementary motion detector. b) Bidirectional motion detector.

Linear lateral inhibition, which was originally proposed by Hartline and Ratliff as a result of experiments on the compound eye of *Limulus*, could explain some of the behaviors of the visual function [Ratliff and Hartline 74]. However, in order to be able to model directional selectivity for motion detection nonlinear interactions are required. The multiplicative lateral inhibition model, which is also known as shunting inhibition, was introduced to provide directional selectivity, and also to describe observations regarding the variation of the shape and size of the receptive fields with changing light intensity [Pinter 84, Bouzerdoum and Pinter 89]. The structure of the shunting inhibitory motion detectors is the same as the EMD cell shown in Figure 2.1.

Structurally different biological motion detectors often add temporal, spatial, or spatio-temporal filters (mainly band-pass filters) before applying the input to the EMD cells [Courellis and Marmarelis 90], or add spatial or temporal feedback to enhance the adaptability to mean luminance, again before the EMD units [Beare and Bouzerdoum 96]. Figure 2.2 shows some of these configurations.

The *template model*, which we have used as the basis for implementation in our chips, is also structurally similar to the EMD cell (see Figure 2.3). However, it only uses digitized values of the temporal contrast. All signals after the thresholding stage are digital values.

Templates can be regarded as low-level spatio-temporal features, in which the motion information is embedded. From the 81 possible templates, eight of them reliably code motion direction under noisy conditions [Nguyen 96]. These reliable templates can be searched for and tracked. In fact, from the point where templates are formed, the motion detection algorithm changes to a feature-based tracking technique.

All these models suffer from the fundamental problems related to local motion detectors, i.e. the *aperture* problem and spatio-temporal aliasing. Biological systems, of course, have much more elaborate mechanisms to overcome these problems, often by integrating motion information at higher processing levels. However, even these simple models pose serious limitations on the hardware resources available in VLSI technologies.

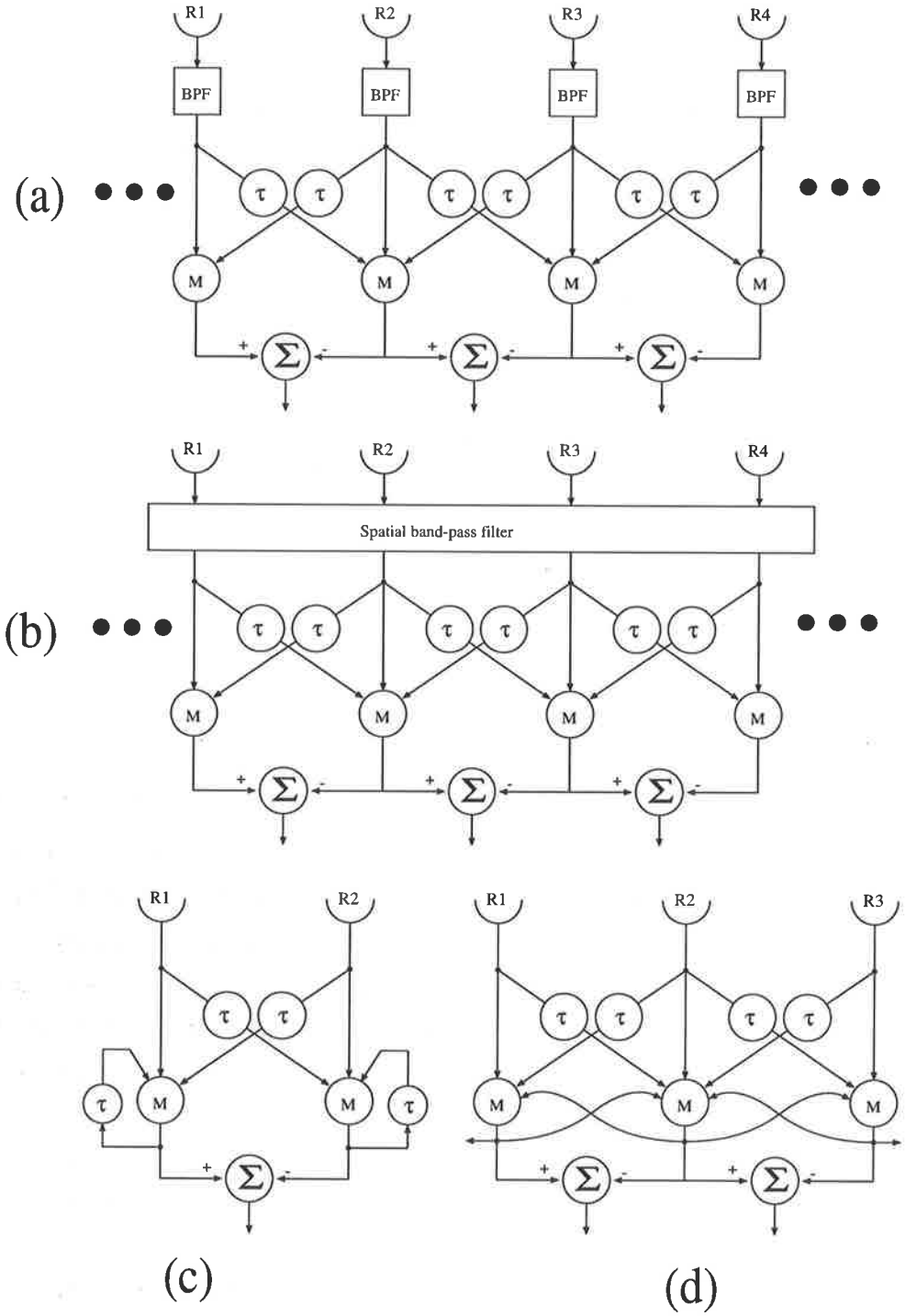


Figure 2.2: Structural modifications to the EMD a) using temporal band-pass filter to eliminate the background illuminance in temporal domain, b) using spatial band-pass filter to remove the spatial background illuminance, c) using temporal feedback, and d) using spatial feedback.

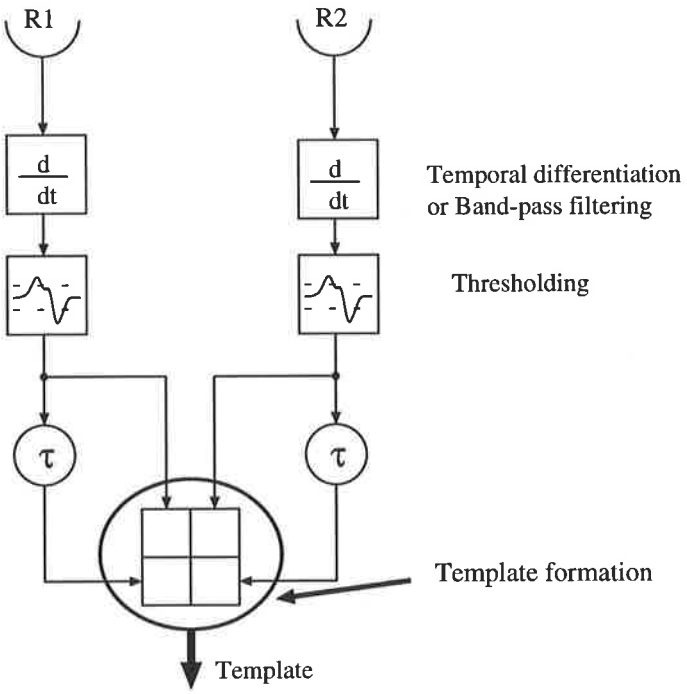


Figure 2.3: The structure of the template model.

2.3 Contrast Enhancement

Contrast enhancement is one of the earliest spatial processing operations in biological vision. Its task is to emphasize interesting features in the image, reduce noise, and make best use of available communication bandwidth by reducing the dynamic range of the image.

In computational vision, contrast enhancement often appears in the context of edge detection and data recovery from a noisy input image. In biological vision, contrast enhancement has been studied under the topic of contrast sensitivity. In most of these studies spatial frequency response is used instead of spatial response. One reason has been that probing many retinal or cortical channels proves impractical. Instead one can provide sinusoidal gratings (or other types of frequency selective stimuli) as the input and measure the response of a single channel, while changing the spatial frequency of the grating.

The function of contrast enhancement can be seen in Figure 2.4-a, which depicts a typical output for a spatial step input. The output level for areas where the input is flat is constant. Also the contrast of the edges is preserved, and depending on the characteristics of the function the contrast may even be amplified. This can be considered as a kind of spatial derivative operation. Another important feature of contrast enhancement is the compression of the dynamic range, as shown in Figure 2.4-b. The smaller dynamic range at the output is necessary for further processing stages, so they do not need to cope with a wide ranging input. The last feature of contrast enhancement is removal of high frequency noise. If the derivate operator is applied to

high frequency noise, the output will be flooded by the amplified noise.

Therefore, contrast enhancement can be seen as a stage for reducing the information by removing the "DC" level, amplifying interesting features (which in this case are spatial frequencies other than low and high-frequencies), and reducing noise by limiting the bandwidth. These properties can be synthesized using a spatial band-pass filter, although we will see that these rather simple characteristics find their roots in more solid mathematical and visual information processing theory.

In the next section some of the most common models for contrast enhancement are described.

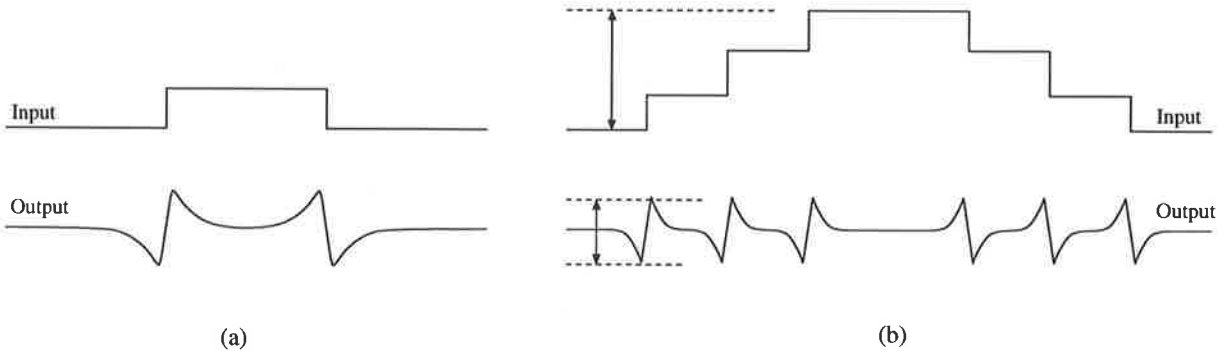


Figure 2.4: The effect of contrast enhancement on step edges.

2.3.1 Computational Models for Contrast Enhancement

Subtraction from spatial average

Intuitively, the subtraction of a signal from the local spatial average, as depicted in Figure 2.5, should yield the same spatial response as in Figure 2.4. The spatial average in fact computes the mean luminance, or in other words an estimate to a signal in a noisy environment [1], from the information in a local neighborhood.

Subtraction essentially removes the local average from the signal and produces a zero-centered output. It is easy to see that if we consider the spatial averaging as a spatial low-pass filter, the whole operation can be regarded as a high-pass filter.

Division by spatial average

In a slightly different approach one can use division instead of subtraction to achieve similar results. Theoretically, there is no apparent advantage, except that by using division the output will be normalized to "one" instead of being centered around zero. If the denominator becomes

[1] This is the basis for the "predictive coding" perspective as the function of the retina [Srinivasan et al. 82]

zero the output will be undefined. However, from a practical point of view this does not pose a serious problem [2].

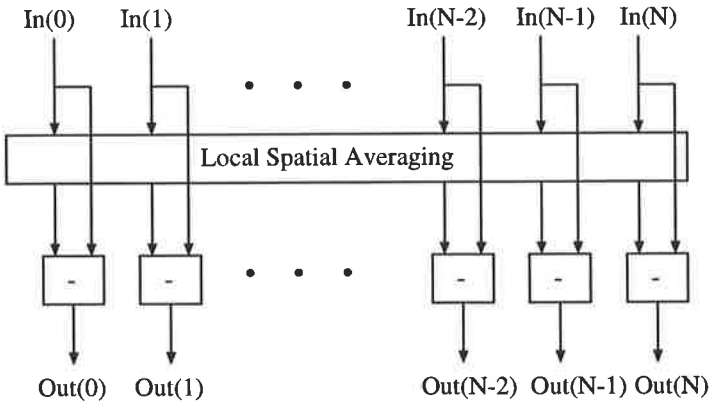


Figure 2.5: Contrast enhancement using division by spatial average.

Laplacian of Gaussian (LoG)

The LoG operator is a process used mainly in edge detection. Gaussian smoothing is known to minimize the uncertainty between the spatial and spatial-frequency spread of the filter. The assumption behind this concept, which was originally proposed by Gabor and later used by Marr and Hildreth, is that edges are localized both in space and frequency, and therefore the edge detection filter should also be localized in space and frequency [Gabor 45, Marr and Hildreth 80]. The Gaussian and LoG operators for a one-dimensional case are given by:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$
$$LoG(x) = \nabla^2 G(x) = -\frac{1}{\sqrt{2\pi\sigma^3}}(1 - \frac{x^2}{\sigma^2}) e^{-\frac{x^2}{2\sigma^2}} \tag{2.5}$$

The parameter σ determines the spatial span of the operator. Two other similar operators, which are more practically oriented, are the difference of Gaussians (DoG) and the difference of exponentials (DoE) described by

$$DoG(x) = \frac{1}{2\pi\sigma 1^2} e^{-\frac{x^2}{2\sigma 1^2}} - \frac{1}{2\pi\sigma 2^2} e^{-\frac{x^2}{2\sigma 2^2}}$$
$$DoE(x) = Ae^{-\alpha|x|} - Be^{-\beta|x|} \tag{2.6}$$

[2] A circuit implementing this model has been used for the MNC operation, which is described in Chapter 8. Due to circuit nonlinearity and saturation, the problem of division by zero is eliminated.

where $0 < \alpha, \beta < 1$ and $A, B > 0$. Both of these functions can be realized using resistive networks (see Section 5.3.2 and Appendix B).

Even Gabor functions

Gabor functions, similar to the Gaussian operator, have the property that the product of their uncertainty in the spatial and spatial frequency domain is minimum. The even and odd Gabor functions have been observed in the cortical receptive fields [Regan 91]. Even (as opposed to odd) Gabor functions also satisfy the few requirements for contrast enhancement, i.e. removal of the DC level and edge enhancement. The first order Gabor function is given by

$$GAB(x) = Acos(\alpha x)e^{-\frac{x^2}{\sigma^2}} \tag{2.7}$$

Regularization theory for surface reconstruction

The problem of surface reconstruction from noisy data is an ill-posed problem, which can be solved by regularizing the problem through proper constraints. In the case of surface reconstruction the following term should be minimized

$$||(A.u - v)||^2 + \lambda||(u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2)^2|| \tag{2.8}$$

where $v = Au$ represents the ill-posed problem, λ is the regularization parameter, and u and v are the input data and the estimated data. The solution to this problem is the *biharmonic* equation [Terzopolous 84].

$$\lambda\nabla^2\nabla^2 v(x,y) + v(x,y) = u(x,y) \tag{2.9}$$

For a one-dimensional case the impulse response can be found as

$$\begin{aligned} BIH(x) &= \frac{1}{2\sqrt[4]{\lambda}}(cos(\frac{|x|}{k}) - sin(\frac{|x|}{k}))e^{-\frac{|x|}{k}} \\ k &= \sqrt[4]{4\lambda} \end{aligned} \tag{2.10}$$

2.3.2 Biological Models for Contrast Enhancement

Linear lateral inhibition

Lateral inhibition as a model for retinal function was proposed by Hartline and Ratliff. Most of the work has been encapsulated in [Ratliff and Hartline 74]. In linear lateral inhibition (LLI)

each cell inhibits its neighbor(s). A network of LLI cells can be described by

$$x_n - au_n - b(u_{n-1} + u_{n+1}) = 0$$
$$\mathbf{X} - A\mathbf{U} = 0$$

$$A = \begin{pmatrix} a+2b & b & 0 & \dots \\ b & a+2b & b & \dots \\ 0 & \ddots & \ddots & b \\ \dots & 0 & b & a+2b \end{pmatrix} \tag{2.11}$$

$$\mathbf{U} = A^{-1}\mathbf{X}$$

where x is the input and u is the output. a and b are the network parameters. Here we have excluded the dynamic part of the original LLI equation, which is a du_n/dt in the right hand side of the equation. One salient feature of lateral inhibition is in the local "interaction", which makes it more attractive from the implementation point of view.

Nonlinear lateral inhibition

Although LLI satisfies most of the requirements for contrast enhancement, it does not explain some other observations, such as the adaptation of the receptive field size and shape to light intensity [Pinter 84]. To extend the functionality of the LLI, it was suggested that nonlinear interactions be used. Multiplicative inhibition (which is known as shunting inhibition) is a simple form of nonlinearity that has been proposed for this purpose [Pinter 84]. The steady state (without the dynamic terms) expression of shunting inhibition is given by

$$x_n - au_n - bu_n \times (u_{n-1} + u_{n+1}) = 0 \tag{2.12}$$

This equation can be linearized and solved relatively easily. The static and dynamic properties of nonlinear lateral inhibition have been extensively studied [Bouzerdoum and Pinter 92,Nabet and Pinter 91].

2.3.3 Comparison

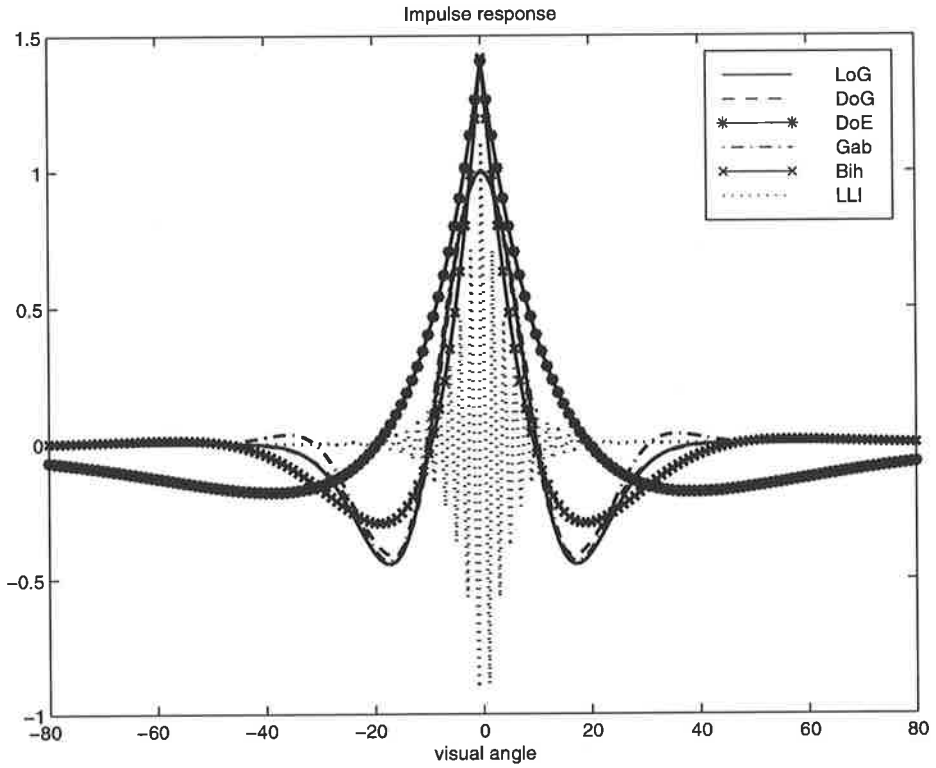
All the models described here can reproduce some of the functionalities of the retina. Which one is the *golden* solution to the widely known properties of contrast enhancement? Figure 2.6-a illustrates the receptive field of the models described. In order to have a fair comparison we have used the normalized LoG operator and have tuned the parameter/s of other models to minimize the mean square error. The subtraction from local average, division by local average, and LLI (with only nearest neighbor interaction) have a span of "one" pixel in the central part of the kernel. In this sense comparing them with other models in unfair.

The LoG, DoG, and the even Gabor functions are very similar in performance. The band-pass filter characteristics of all models is clearly observed in the frequency response. If we had tuned the parameters of each model such that the peak frequencies were the same, it could be seen that the solution of the biharmonic equation, and the DoE (difference of exponentials) have a slightly higher high-frequency signal transfer. The even Gabor function, by contrast, has higher low-frequency signal transfer.

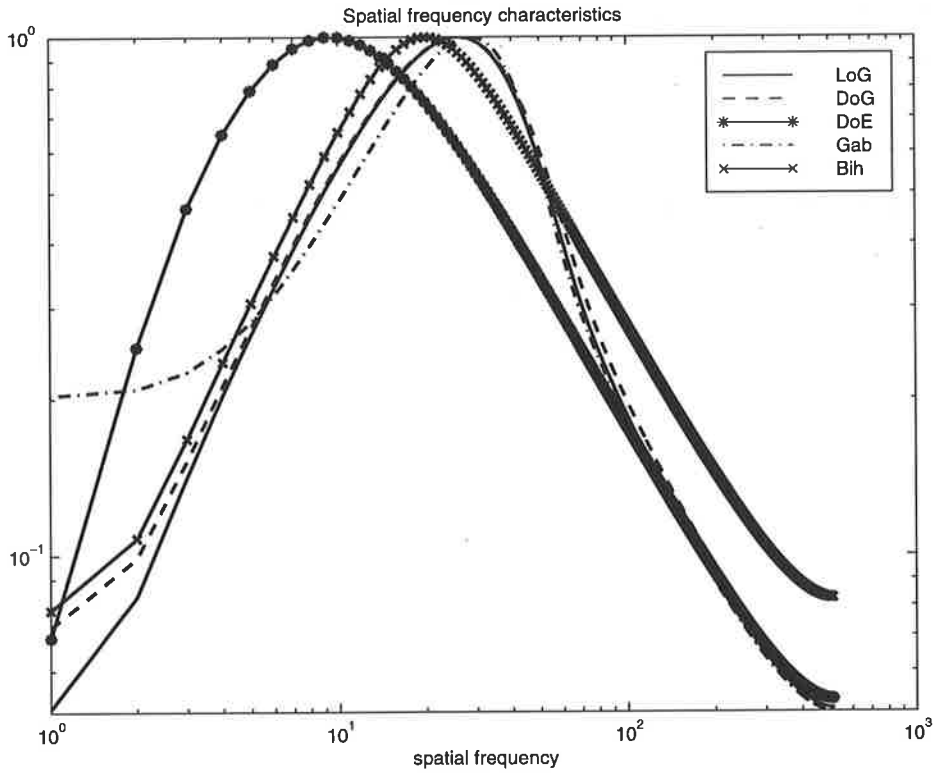
As illustrated in Figure 2.7, typical contrast sensitivity for the retinal and cortical layers resembles that of the Gabor function. Consequently, most of the literature on the subject of modeling contrast sensitivity and characterizing receptive fields in the retinal and cortical layers use Gabor functions (see for example [Watson and Solomon 97]).

From a VLSI point of view, realizability is regarded as more important than trying to copy the exact shape of the receptive field. The accuracy of the implemented circuits is usually far less than what is needed to produce a distinguishably different receptive field. Therefore, unless the very detailed characteristics of the model are crucial to the implementation, the general characteristics of contrast enhancement can be replicated by any of the models described here.

Implementing any function that involves a Gaussian operator, for example in the DoG or Gabor functions, requires negative resistive connections to the second nearest neighbors [Kobayashi et al. 91]. This requires large circuits, particularly because two separate Gaussian smoothing networks should be used. Second-order resistive networks have also been used to implement Gabor-like (an exponential modulated with cosine or sine) functions [Raffo 95]. A very compact implementation, which approximates the solution of the biharmonic equation has been described in [Andreou and Boahen 94, Boahen and Andreou 92]. Section 5.3.5 describes the circuits and networks for contrast enhancement.



(a)



(b)

Figure 2.6: a) Spatial receptive fields of different contrast enhancement models. b) Spatial frequency characteristics.

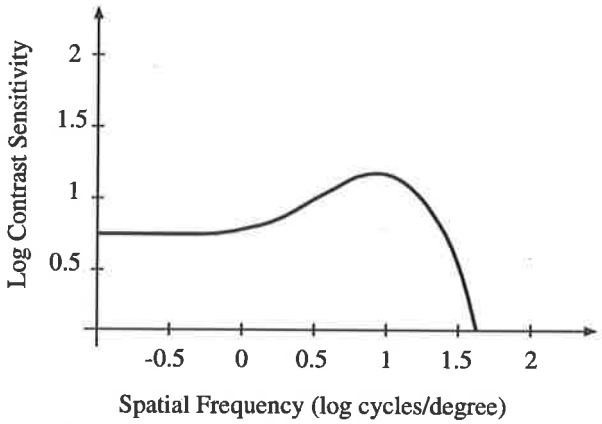


Figure 2.7: Spatial contrast sensitivity function for an average foveal primate lateral geniculate nucleus (LGN) neuron [Watson 91].

Chapter 3

Analog VLSI for Computer Vision

"There is nothing that is done in the nervous system that we cannot emulate with electronics, if we understand the principles of neural information processing"

Carver Mead in IEEE Proceedings, 1990

This chapter gives another perspective of the project, i.e. from the hardware point of view toward algorithmic issues. First the general issues facing AVLSI design of vision algorithms are presented. Analog neural networks, as an architectural superset of vision chips, are then introduced. Major differences between analog neural networks and vision chips are highlighted.

The advantages and disadvantages of VLSI technologies for implementing vision chips are discussed. Dominant design techniques for vision chips, i.e. current/voltage mode, charge mode, and hybrid modes (mixed digital/analog, and mixed charge/voltage), are presented.

Introduction

Although the algorithm development stage is an essential part in the design of any artificial vision systems, hardware implementation issues play an equally important role when real-time[1] operation of the system is considered. The first limitation in realizing real-time hardware, which is almost independent from the algorithm being implemented, is the information transfer bottleneck from the imager to the processor. For a 1024×1024 pixel image, digitized at 8-bit and 30 frames/second, a data transfer bandwidth of about 30 M-Bytes/second is required. In 33 seconds there will be about one Giga byte of image data. Even the storage and retrieval of this amount of information using current high-end workstations is not feasible in real-time. Of course, the amount of information is related to the image resolution, frame rate, and processing requirements of the algorithm. These parameters are all dependent on the application and the algorithm. However, for real-time applications this information should be processed at the same rate as the input.

Therefore, the first step towards achieving real-time performance is to reduce the amount of information transferred from the image acquisition to subsequent processing stages. Parallel processing architectures present another obvious solution. Many vision chips in fact exploit both concepts by incorporating one processor for each pixel, and at the same time reducing the amount of information. Some vision chips target only the information reduction by performing image compression [Aizawa et al. 95, Hamamoto et al. 96, Hamamoto et al. 97].

All computer vision algorithms can be implemented in serial digital hardware, as long as issues such as speed or size are of no concern. This is simply because there is no constraint on making the structure and flow of data in the model and the hardware/software architecture the same. Parallel hardware architectures constitute a step forward in realizing some of the natural parallelism in computer vision problems. However, they have been severely limited by problems such as data partitioning, as the size of the problem is often larger than the size of the processing elements in the hardware, and rewriting algorithms to suit the specific hardware. Nonetheless, all vision algorithms can be implemented in general purpose parallel hardware, although the efficiency of the implementations may not be optimal.

Analog implementations of vision algorithms can be divided into two main categories: those using general purpose analog processing networks (mainly neural networks), and vision chips. Analog neural networks retain the generality of neural networks, while offering a high level of parallelism in a small area. Vision chips, on the other hand, represent the ultimate architectural similarity between the algorithm and the hardware.

In this chapter, various topics relevant to VLSI design issues are presented. The purpose is to present a view of the requirements for VLSI implementation of vision chips, and the capabilities of current technologies and design methodologies in addressing these requirements.

[1] Here we call a process real-time if the processing rate keeps pace with the information rate at the input.

3.1 Analog or Digital?

A philosophical point

An important issue in the design of any hardware system is the choice of analog or digital implementation in various parts of the system. Before presenting a more technical discussion, we would like to highlight a rather philosophical point. In designing a system, if only **engineering** aspects of the implementation (such as power, area, speed, robustness, and so on) are the main concern, then one cannot elaborate on ideas which are not realizable using available technologies, design techniques, and knowledge. However, if the design is performed under a **scientific** (versus engineering) framework, there will be freedom in exploring new ideas, techniques, and so on. These are simply the definitions of engineering and scientific work. For example, it is clear that if technology permitted, we would have been able to build silicon retinas which could outperform conventional camera-processor systems for vision processing. There is no need for a hard proof other than referring to any biological vision system. However, the technology at our disposal is far inferior to its biological counterpart. Our attempt at understanding and emulating biological vision systems, therefore, should be considered from a scientific rather than an engineering viewpoint. At this stage, we do not expect to build vision chips that can achieve significantly higher performance. But as this area progresses, and the technology becomes available, more mature engineering designs would be possible.

Design trade-offs

A general comparison between analog and digital implementations would be meaningful only in the context of a specific design. In fact, most comparisons performed to date have been performed in relatively narrow fields. This is simply because the design trade-offs and costs are well understood, and a decision can be made based on particular implementations (For example see [Hahm and Titlebaum 97, Sarpeshkar 97, Hosticka 85]).

Some of the main differences between analog and digital implementations are listed here.

- **Signal representation:**
  In analog implementations, signals are represented by continuous levels, while in digital, signals can only take two quantized levels. In order to represent more levels, more bits are required in the form of a *word*. A word can represent an integer, a floating point number, or any other type.

- **Computation:**
  Analog systems can effectively use the physical principles of a device to perform some computations. For example addition or subtraction in current-mode circuits only requires

a connection between the wires. However, this depends on the design style and techniques. Addition using voltage-mode design techniques requires more elaborate circuits. The concept of *physics of computation* highlights the computational efficiency of analog circuits. Using physical characteristics of the devices we may be able to perform operations in a very efficient way.

In digital systems every operation consists of many single boolean operations.

- **Dynamic range:**
  In analog implementations, dynamic range is limited by several factors, including supply voltage, and noise (thermal, interference, and distorsion). The typical dynamic range of analog signals is in the order of 40 to 100 dB.

  The dynamic range of a digital word is limited by the number of bits and the type of the word. A 32-bit word can have a dynamic range of 200 dB in an integer representation or a dynamic range of more than 800 dB using a floating point representation.

- **Noise:**
  Noise in analog systems arises from stochastic processes, such as thermal and shot noise, or deterministic processes, such as interference and distortion. In a cascade of analog circuits, noise is accumulated at each stage, and eventually may completely dominate the signal.

  In digital systems, noise is virtually removed at each low-level gate, as almost all digital gates regeneratively recover the signal. Therefore, the only source of noise in digital systems are the quantization noise, and round-off errors.

  A similar regenerative process may be employed in an analog system. This is known as A/D/A (analog to digital to analog). However, the associated cost of the extra circuitry makes this approach suitable only for very low precision analog processing (at most 5 bits).

- **Biological plausibility:**
  Biological systems use analog signals for transmitting neural information. Pulse coded signals are also seen in long axonal signal transfers. Pulse code modulation is a mixture of analog and digital representations, where only one wire is used to represent an analog signal (coded in the frequency or the width of pulses) using quantized levels. It is well known that pulse coded signals are more robust against noise and distortion than continuous analog signals.

- **Speed:**
  Speed of processing in both analog and digital circuits is directly proportional to the

required dynamic range. The power is also directly related to speed. Therefore, there is not usually a significant difference between the power-delay products of analog and digital implementations, with the same dynamic range and using the same technology.

- **Area:**
  If the physical properties of the device are used in the implementation, the area for implementing an analog function could be about two orders of magnitude less than its digital counterpart. However, as the functionality and precision of the processing increase, conventional analog circuits do not provide a clear advantage, and often for precisions of more than 8 bits the area occupied by an analog implementation becomes comparable to that of a digital implementation.

  It should be emphasized again that design style (or design ingenuity) significantly affects the area. As an example see Section 5.3.3, which describes various implementations of spatial smoothing. One design uses only two transistors per pixel, while another design uses 12 transistors.

As in every system, selecting a particular approach depends on the design requirements. In vision chips, area is the dominant factor. Relatively complex functions must be implemented to perform even simple tasks. To obtain vision chips with enough resolution, it is necessary to minimize the area occupied by the processing elements at each pixel. As the process technology scales down, more circuitry can be placed at each pixel, and the resolution can be increased.

With the currently available technologies, it is not feasible to implement vision chips using digital implementations with usable resolution, and with the same natural architecture as biological vision systems. The area required for implementing digital functions, such as multiplication is very large, even in the leading edge processes. To this we should add the memory and interconnection requirements, and the analog to digital conversion stages for interfacing to the sensors. Although a digital vision chips exists [Bernard et al. 93], that particular implementation operates on binary images and uses boolean operators. This might be satisfying for specific applications, but cannot in general be a substitute for the requirements of gray-level images.

For neural networks, the requirements are slightly less restrictive than vision chips. Vision chips comprise photodetectors and processing elements and often there is a close interaction between the two. Also, the input is a two-dimensional image and ideally there is one processing element for each photodetector. In neural networks, the input is not necessarily two-dimensional. Also, feedback paths are an indispensable part of any neural network with learning capability. A simplified illustration of a vision chip and a Hopfield neural network is shown in Figure 3.1. Neural networks are mainly limited by the interconnection density and the synapse circuit, which basically multiplies the weight by the input (and output) and can be constructed using simple analog circuits in a small area.

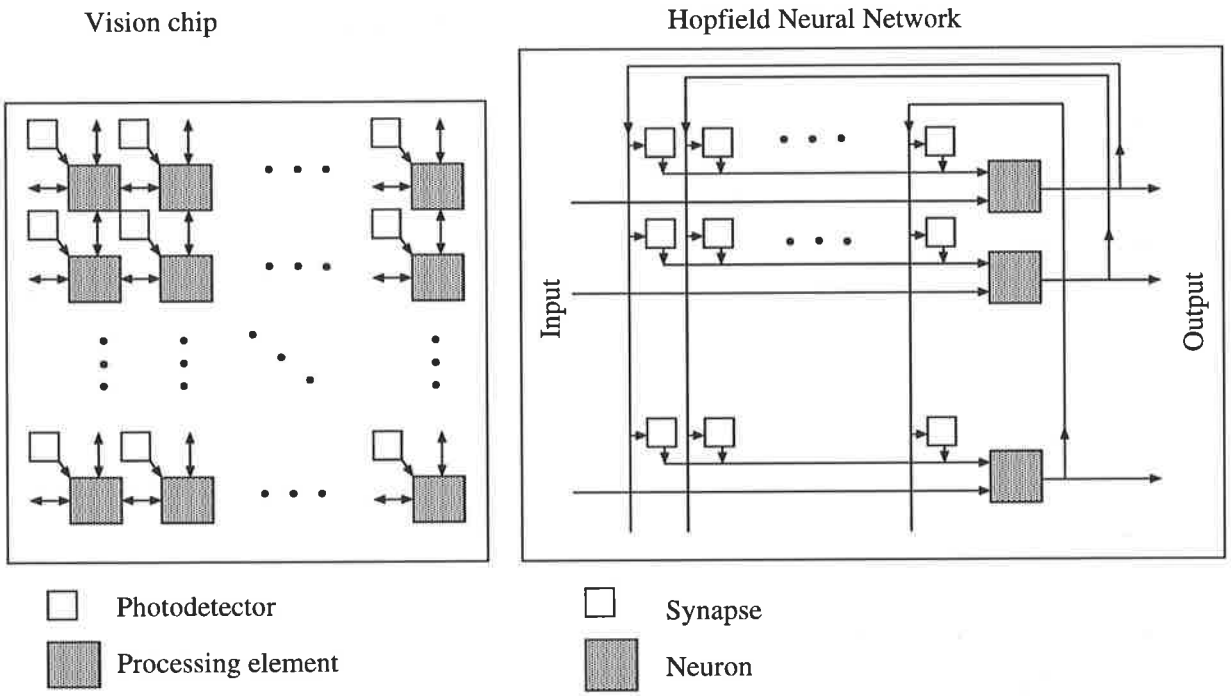


Figure 3.1: A general view of a vision chip and a neural network.

3.2 Analog VLSI and Artificial Neural Networks

The hardware implementation of artificial neural networks (ANN) has been researched extensively, and several design methodologies have already been adopted. There are close architectural similarities between vision chips and ANNs, and most models for vision can be categorized as a subclass of neural networks. Therefore, some of the neural net design techniques may be used for vision chips. The purpose of this section is to give a quick overview of some of the common architectures and their VLSI implementation requirements, rather than to elaborate on the theoretical or architectural aspects of neural nets, as there is a large amount of literature available on these subjects.

Some of the most common neural net architectures are illustrated in Figure 3.2. An important observation, is the massive number of interconnections required in all of these architectures. This has been a major barrier in designing large neural networks. In a general implementation a fully connected network is required, although in some cases the interconnections may be sparse and limited to a local neighborhood. For networks with error back-propagation, the number of interconnections doubles and each processing node should also include an error calculation module. Implementing any of these architectures, except for cellular neural networks (CNN), is virtually infeasible using current technologies for a *reasonable size two-dimensional input vector*. It is therefore not surprising that none of these architectures have been mentioned in any of the vision chips reported to date. CNNs, on the other hand, have a VLSI friendly structure

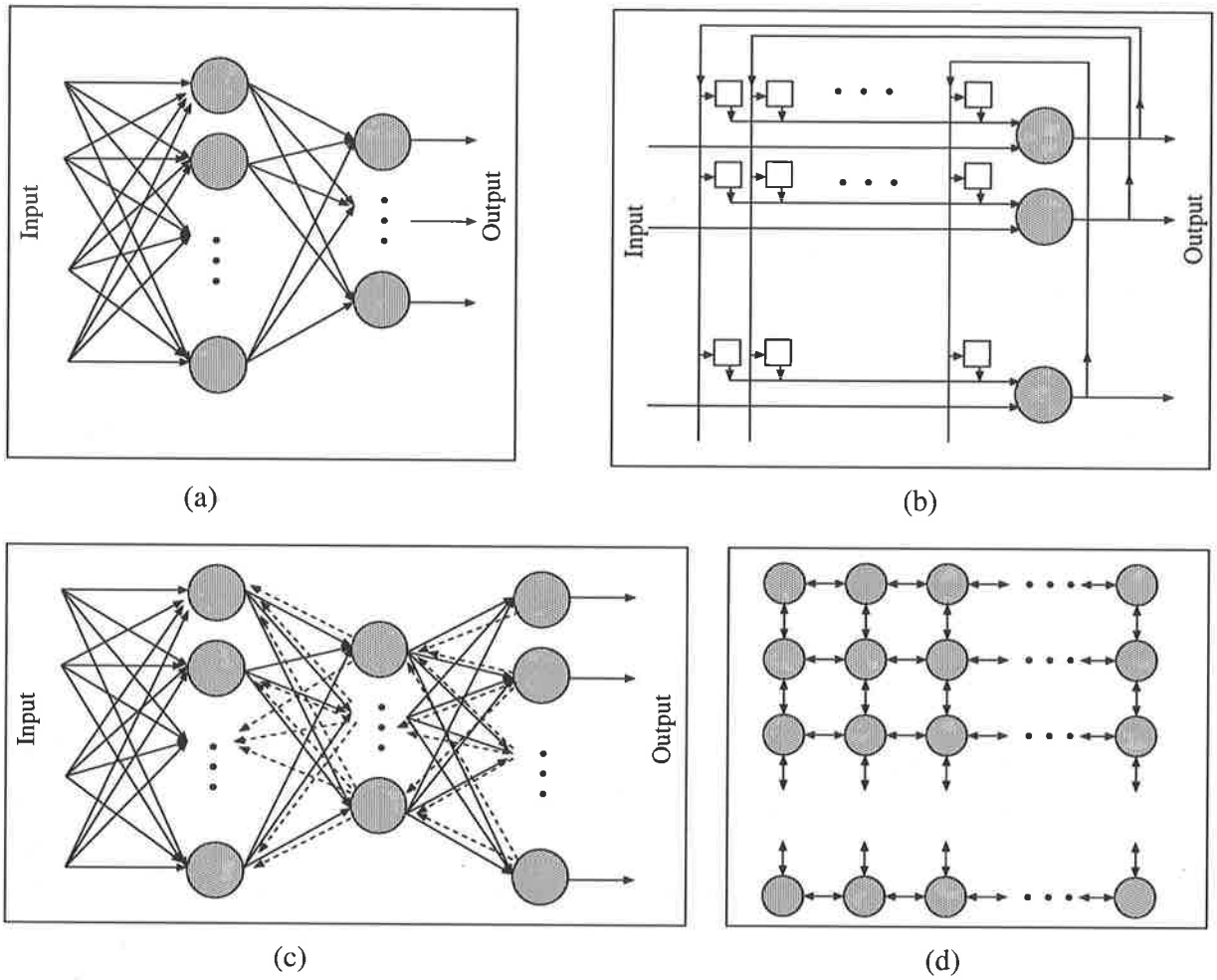


Figure 3.2: Some neural network architectures. a) A feed-forward network. b) a fully connected Hopfield network. c) a multilayer perceptron (MLP) with error back-propagation (BP). The dashed arrows in MLP indicate the flow of error signals. d) a cellular neural network (CNN)

and have been used in the implementation of several vision chips (see Sections 5.3 and 5.3.1 for some of the applications of CNNs for this purpose.).

Neural networks are often characterized by an **interconnected** network of a **large number** of **simple processing elements**, which may or may not have **feedback**. In general, for a fully connected single-layer one-dimensional neural network, $O(N^2)$ weight multiplication and storage elements are needed. The four highlighted points in the definition of a neural net, and the weight storage requirement, demonstrate the demands of a NN.

Interconnection

Interconnection is the first requirement of a NN, which poses a major limitation in a hardware implementation [Ismail and Fiez 94]. Real life applications require a large number of neurons to be connected together. Routing a large number of wires inside a chip can hardly be achieved for a NN with several hundred neurons. As the processes shrink, the interconnect density increases, but at the same time the neuron density increases at a rate proportional to the square of the neuron numbers. In addition, the scaling criteria usually tend to scale the interconnects less than other device dimensions, to reduce the interconnect resistance. Hence, it is necessary to use multiple interconnect layers. Today's advanced VLSI processes have up to four layers of metal interconnection. It is expected that VLSI processes with six layers of interconnects will become available by the year 2000 [Davari et al. 95].

The number of neurons

The number of neurons implementable in a single chip is limited by the available area. In one-dimensional neuron arrays the number of neurons is mainly limited by one dimension of the chip, and the circuits can extend in the other dimension. In two-dimensional arrays, such as 2-D vision chips, the number of neurons and their size are limited in both directions. In advanced submicron processes the number of neurons in each direction can only grow to a size of 100 to 200. 2-D NN chips are also limited by the required 2-D interconnection. Unless a 2-D signal transfer method are used, such as an optical input and output interface [Nitta et al. 95], or a 3-D structure [Kioi et al. 92, Crowley and Vardaman 94, Larcombe et al. 95, Stern et al. 96], the information can only be transferred serially.

Processing elements

The processing elements used in a NN include the neuron activation function, weight multiplication, and weight update blocks. The activation function, which is usually the sigmoid function, can be implemented easily using continuous or piecewise linear circuits. As the number of neurons is usually an order of magnitude less than the number of weights, the activation function is implemented using more general designs. The number of weight multipliers is a $O(N^2)$ function for a 1-D neural net. Therefore, special attention should be paid to the size of the multiplier.

Weight storage

Weight storage is another major obstacle in achieving large VLSI neural nets. For a 1-D neural net $O(N^2)$ weight memories are required. Various methods have been used for storing synaptic weights [Mead and Ismail 89, Moon et al. 92, Linares-Barranco et al. 93, Ibrahim and

Zaghloul 90, Inigo et al. 90, Morishita et al. 90]. Design aspects for synaptic memory include weight accuracy, storage time, storage method (analog or digital), and synapse size. Weight accuracy in most analog implementations, whether the weight storage is digital or analog, is between 6 and 8 bits. Storage time for analog storage methods is limited by the leakage current at the storage node. The gate capacitance of MOS devices has been used as a dynamic analog memory, with a short storage time and low accuracy. Floating gate transistors can store charge for a longer period with almost the same accuracy as dynamic memory elements.

Feedback or learning

Most neural networks require some form of learning. Therefore, the learning capability of a neural net chip significantly improves its system level integration [Tawel 93]. The implementation of different algorithms on a chip is limited by the accuracy of the computation. The back-propagation algorithm, for example, requires high accuracy to stabilize. The accuracy requirement, in general, depends on the learning algorithm and the specific problem.

There are several analog implementations with on-chip learning capabilities [Hirotsu and Brooke 93, Linares-Barranco et al. 93, Nitta et al. 92a, Hammerstorm 90, Hochet et al. 90, Macq et al. 93, Salam and Choi 90, Tam et al. 90, Wang et al. 93]. Most digital implementations of neural nets have embedded facilities to implement different learning algorithms as they do not often have a large number of neurons [Kondo et al. 94, Sato et al. 93, Oteki et al. 93, Nakahira et al. 93, Morishita et al. 93, Fujita et al. 93, Yasunaga et al. 93].

3.3 Analog VLSI and Vision Chips

Although vision chips implement some form of neural network, and their function can be formulated in a similar fashion to neural nets, the physical VLSI limitations have had a significant effect on the topology, architecture, and functionality of vision chips.

Here some of the main VLSI requirements and limitations of vision chips are highlighted.

- **Photodetection:**
  The input to a vision chip is provided by photodetectors. Trade-offs in the design of photodetectors necessitate a large allowance for their area. In theory, the dynamic range of the photodetectors determines the ultimate dynamic range of the system. Fortunately, most photodetectors available in standard fabrication processes have a dynamic range of at least five decades. In practice, the photocircuits, which transduce the photocurrent into a voltage, reduce this dynamic range.

- **Interconnection:**
  Due to the two-dimensional nature of vision chips, interconnection is limited to the first, and in some cases to the second nearest neighbors. This in effect dictates the type of algorithms which can be implemented in vision chips.

- **Learning or adaptation:**
  Learning in the same sense as neural networks is very difficult to implement in vision chips, because of area limitation. However, adaptation, which can be considered as a lower-level learning mechanism, may be implementable. Adaptation can, for example, change the gain of transduction, or reduce offsets and mismatches in transistors. Learning, in its general sense, is possible only with extra computational modules to calculate the error from the network and feeding it back to correct the network parameters.

- **Area:**
  As emphasized before, area is the most limiting factor in realizing vision chips. The following function roughly quantifies the cost per unit area.

$$Area = \alpha N_T^2 \times T_A + \beta N_C^2 + \gamma N_B + P_A \tag{3.1}$$

  where N_T is the number of transistors, T_A is the average area of transistors, N_C is the number of interconnection to and from a pixel at one side of the pixel, N_B is the number of bias, control, and read-out signals required to operate the circuit, and P_A is the photodetector area. α, β, and γ are constants.

  The contribution of the P_A is often less than 25%, and may be as low as 5%. The quadratic term for the number of transistors, N_T, is a result of placement and routing areas for the

transistors. The term T_A accounts for transistor sizing, for example to control the level of mismatch.

- **Computational devices:**
  With the extensive design techniques and circuits available for analog design, most computational functions can be easily realized. For example translinear circuits can compute any function of the form

$$Out = \frac{\prod_n (I_{n0} + I_n)^{a_n}}{\prod_m (I_{m0} + I_m)^{b_m}} \tag{3.2}$$

  where I_{n0}, I_{m0}, a_n, and b_m are constants. Many other functions, such as trigonometric functions, can be approximated using translinear circuits [Seevinck 88].

  In the framework of *physics of computation* it would be desirable to perform each function using the natural physics of a single device. This is of course only possible for a very small set of operations, such as addition (current summation), and logarithm and exponential functions (the I-V characteristics of a diode or a bipolar transistor). Biological systems, on the other hand, create a special *device* for each *function*. By using devices with fixed functionality, complex functions can be implemented by combining several devices, and this is where the art of circuit design comes in.

3.4 Technology Base for Implementation

Choosing a specific technology for implementing a vision chip depends on the design requirements, and the capabilities of that technology. The major VLSI design resources required for implementing vision chips were described in the previous section. In this section the dominant VLSI technologies are reviewed. In addition to general comments, the equations which characterize the basic devices are presented in order to highlight their functionality. Device physics, and detailed equations concerning second order effects, are not discussed here.

3.4.1 CMOS

CMOS has been and will remain the dominant VLSI technology for most applications, including vision chips. This is due to the more established design methods and techniques, the widespread availability of CMOS processes, and its suitability to both analog and digital circuit design.

The basic device structure for a conventional MOSFET is shown in Figure 3.3, and the device equations are shown in Table 3.1. In the equations C_{ox} is the capacitance per unit area of the gate, W and L are the width and length of the transistor gate, V_T is the threshold voltage of the transistor, U_T is the thermal voltage (i.e. kT/q), I_{DS} is the drain-source current, V_{GS} and V_{DS} are the gate-source and drain-source voltages, and V_{GB}, V_{SB}, and V_{DB} are the gate, source and drain voltages with respect to the substrate voltage (in this case the well potential).

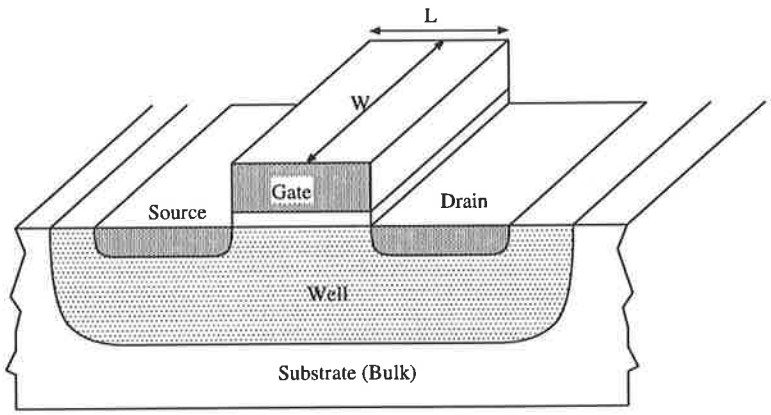


Figure 3.3: A MOS transistor.

In addition to the large range of operations implementable using these equations, a MOS transistor also offers other types of embedded devices. The MOS capacitor, for example, can be used as a storage element with high capacitive density. The gate of the transistor also can be left floating, and used as a long-term storage device. The physics of the device can be used further by storing and transferring charge through potential wells created in the channel of the

Table 3.1: Device equations for a MOSFET.

Region	Equation	Conditions
Strong Inversion	$I_{DS} = \mu C_{ox} \frac{W}{L}(V_{GS} - V_T)^2$	$V_{GS} > V_T$, $V_{DS} > V_{GS} - V_T$ $V_{DB} > -V_0$, $V_{SB} > -V_0$
Linear Region	$I_{DS} = \mu C_{ox} \frac{W}{L}[(V_{GS} - V_T)V_{DS} - \frac{1}{2}V_{DS}^2)]$	$V_{GS} > V_T$, $V_{DS} < V_{GS} - V_T$ $V_{DB} > -V_0$, $V_{SB} > -V_0$
Weak Inversion (Subthreshold)	$I_{DS} = I_0 \frac{W}{L} e^{\frac{V_{GB}}{nU_T}} (e^{-\frac{V_{SB}}{U_T}} - e^{-\frac{V_{DB}}{U_T}})$	$V_{GS} < V_T$, $V_{DS} > V_{DS,Sat} \approx 4U_T$ $V_{DB} > -V_0$, $V_{SB} > -V_0$

transistor. All the junctions in the transistor can be used as photodetectors (see Section 5.1). There are parasitic bipolar devices available, shown in Figure 3.4. A MOS transistor can also be operated as a switch, and hence be used in switched circuit designs.

A CMOS process offers a rich set of devices. Although some of these are parasitic elements, they can be used freely and without any extra cost in terms of the process steps.

3.4.2 BiCMOS

In BiCMOS processes, bipolar as well as CMOS transistors are available. Bipolar devices have been the workhorse of analog circuit design. They can be used to increase the speed of circuits, whether analog or digital, or be used to implement circuits which are not easily or reliably implementable using CMOS devices only. For example, implementing translinear circuits in CMOS requires operating in the subthreshold region. It is well-known that the mismatch of transistors in this region is very high, which is unacceptable for many conventional circuits. Therefore, in cases where device matching is important, bipolar transistors should be used.

Although special BiCMOS processes are required for high quality bipolar devices, parasitic bipolar devices in normal CMOS processes can be utilized, as shown in Figure 3.4. This is in fact very advantageous for analog VLSI circuits, which do not need to operate at high speeds.

Table 3.2 shows the first order device equations in the Ebbers-Moll formulation for a bipolar transistor. In the equations I_E, I_B, and I_C are the emitter, base, and collector currents, respectively. V_{BE}, and V_{BC} are the base-emitter and base-collector voltages. I_{CS} and I_{ES} are the collector and emitter saturation currents. α_F and α_R are the ratios of collected current to

Table 3.2: Device equations for a bipolar transistor.

Region	Equation	Conditions
All regions	$I_E = I_{ES}(e^{V_{BE}/U_T} - 1) - \alpha_F I_{CS}(e^{V_{BC}/U_T} - 1)$ $I_C = \alpha_R I_{ES}(e^{V_{BE}/U_T} - 1) - I_{CS}(e^{V_{BC}/U_T} - 1)$ $\alpha_R I_{ES} = \alpha_F I_{CS}$	$(V_{BE} > 0\ ,\ V_{CE} > V_{CE,sat})$ or $(V_{BC} > 0\ ,\ V_{EC} > V_{EC,sat})$

injected current in the forward and reverse mode of transistor operations.

A large number of functions are realizable using these equations. Although BiCMOS processes offer the extra bipolar transistor, their use has been limited due to the more complex fabrication process, and the increased cost. A conventional BiCMOS process costs around 50% more than a CMOS process with the same feature size.

BiCMOS processes do not offer significant advantages for the design of vision chips. Parasitic devices can serve as an alternative, whenever bipolar transistors are required,

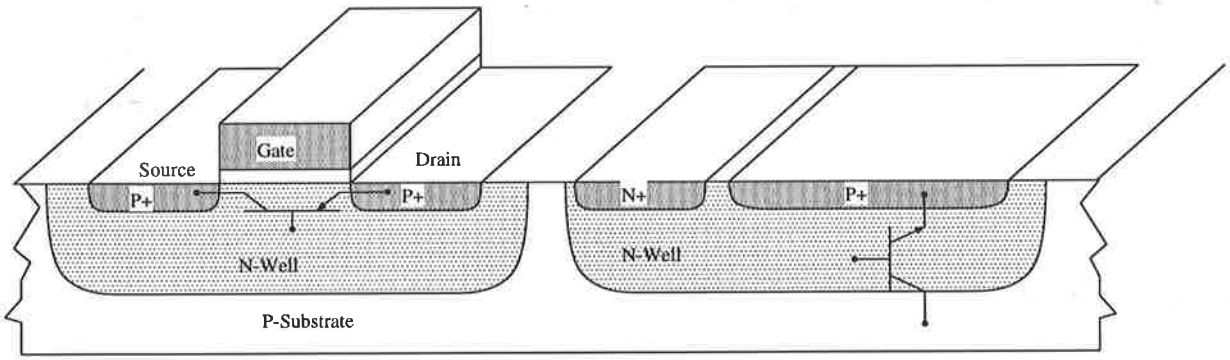


Figure 3.4: Parasitic bipolar transistors in a CMOS process.

3.4.3 CCD/CCD-CMOS

Charge coupled devices (CCD) operate on charge packets in potential wells created by applying voltages to the gate of MOS capacitor structures. Figure 3.5 illustrates the concept. For proper operation, the order and the magnitude of voltages applied to the CCD elements should be carefully selected. CCD processes have several disadvantages that make them unattractive for vision chip applications, despite the fact that CCD imagers are very common and their design has evolved to a high level. These disadvantages are:

- CCD processes are tuned for charge transfer quality. Therefore, MOS transistors available in these processes have very poor characteristics.
- The only operations easily implementable using CCD devices are charge summation and charge subtraction (this is possible by manipulating the depth of the potential wells). More complex functions can only be implemented using CMOS circuits. This necessitates additional circuits for reading and writing charge information to and from the potential wells.
- There is a large number of clock signals required to operate a CCD design, and a large power budget should be assigned for this.

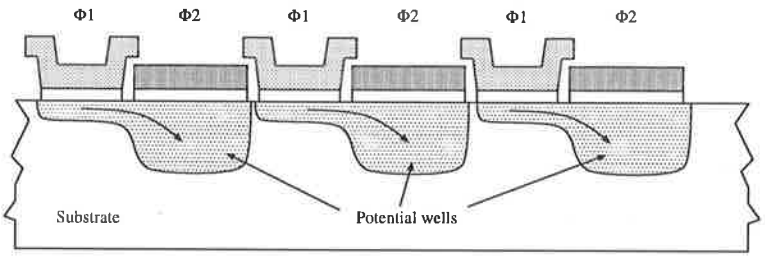


Figure 3.5: CCD structure.

3.4.4 GaAs

Gallium-Arsenide (GaAs) devices, such as MESFET (MEtal Semiconductor FET) and HEMT (High Electron Mobility Transistor) structures, are mainly used for very high speed applications, thanks to the higher carrier mobility of GaAs material. The structure of a MESFET is shown in Figure 3.6, and device equations are listed in Table 3.3. Note that the equations are very similar to those of a MOSFET.

An important feature in MESFET and HEMT devices is that, unlike MOSFETs, the gate is not isolated from the channel, and has a schottky diode connection. Therefore, a large leakage current is expected from the gate. This simple fact has hampered the design of both analog and digital circuits. Also the gate voltage cannot exceed the turn-on voltage of this schottky diode, which means that designing storage elements is impractical or very inefficient in GaAs devices.

The only advantage that GaAs technology may offer is in designing systems which include light emitting devices [Nitta et al. 92b, Nitta et al. 93]. This characteristic may be used for implementing two-dimensional signal transfer between multiple chips.

Overall, the current GaAs processes do not have significant advantages for vision chip design.

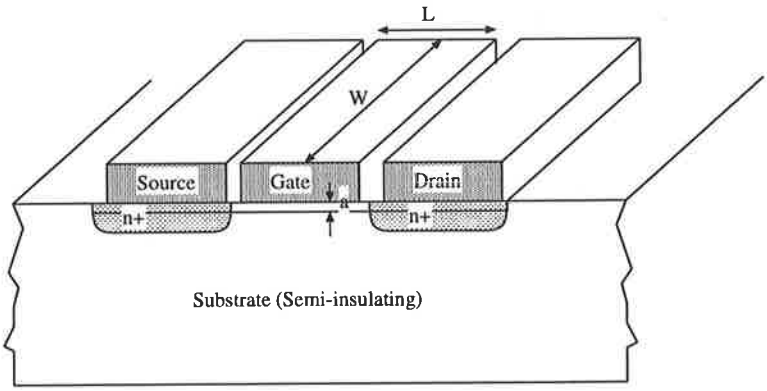


Figure 3.6: The structure of a MESFET.

Table 3.3: Device equations for a MESFET.

Region	Equation	Conditions
Strong Inversion	$I_{DS} = \beta(V_{GS} - V_T)^2$	$V_{GS} > V_T$, $V_{DS} > V_{GS} - V_T$ $V_{DB} > -V_0$, $V_{SB} > -V_0$
Linear Region	$I_{DS} = \beta[(V_{GS} - V_T)V_{DS} - \frac{1}{2}V_{DS}^2)]$	$V_{GS} > V_T$, $V_{DS} < V_{GS} - V_T$ $V_{DB} > -V_0$, $V_{SB} > -V_0$

3.5 Analog VLSI Design Methodologies

Within the boundaries of a specific technology there are several circuit design techniques that can be used in an optimum way. For example, CCD technology is essentially limited to charge-mode circuit design. The conventional VLSI technologies have been used in most of the design methodologies which will be described in the following sections.

From the previous section it is clear that CMOS technology constitutes the best choice, unless the specific features of a particular technology are essential to the performance. Therefore, in the following sections circuit implementations using MOS transistors are focused on. Again attention is paid mainly to the suitability and unsuitability of each technique for designing vision chips. It is assumed that proper background for the analysis and synthesis of each technique exists.

An implementation of spatial smoothing is presented as an example in each section, to provide an idea about the capabilities of each technique.

3.5.1 Current Mode & Voltage Mode

In every circuit, current and voltage coexist. The distinction between a current-mode or a voltage-mode circuit is based on whether the signals associated with the system variables are represented using currents or voltages. Both techniques have been widely used in many applications, and in some circuits both modes are present. For example, in a Gilbert multiplier the two inputs are voltage and the output is a current. Current-to-voltage and voltage-to-current conversion circuits can be used to translate a signal from one mode to the other.

It is widely known that current-mode circuits can present a larger dynamic range. This is only because the elements that process the currents do not heavily depend on the voltage across their ports. A unity gain current mirror, shown in Figure 3.7, can have a dynamic range of more than seven decades. However, this is eventually limited by the voltage at the input and output nodes. In the subthreshold region the input voltage is dependent on the input current by a compressive logarithmic relationship, and in the strong-inversion region by a square root function.

In a voltage mode approach the dynamic range is directly limited by the supply voltage and the achievable accuracy.

One major advantage of current mode circuits is the simplicity of addition and subtraction. In voltage mode circuits special circuits are required to implement these basic functions. Subtraction of voltages is only possible using active components, such as operational amplifiers (Op-Amp). A compact implementation of these basic operations can significantly reduce the area because these are the most frequent operations in many algorithms.

Current mode CMOS circuits operating in the subthreshold region can use the translinear principle to implement a wide variety of functions of the form

$$Out = \frac{\prod_n (I_{n0} + I_n)^{a_n}}{\prod_m (I_{m0} + I_m)^{b_m}} \tag{3.3}$$

Also *externally linear* log-domain filters can be implemented in a small area, and with a small number of components [Seevinck 90, Tsividis 97]. Many other circuits for implementing various functions necessary in vision chips have, in one way or another, used the current mode approach in the design (see Part II: *Synthesis of Vision Algorithms*).

Figure 3.8 shows the implementation of spatial smoothing in current and voltage mode circuits. Notice that in voltage mode either fixed resistors, or variable resistive elements should be used. Fixed resistors are not practical in AVLSI due to area limitations, and resistive elements are bulky.

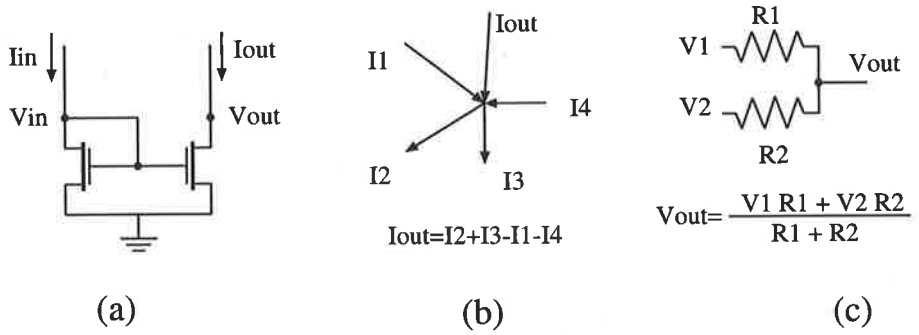


Figure 3.7: a) A current mirror. b) Current mode addition and subtraction. c) Voltage mode addition using resistors.

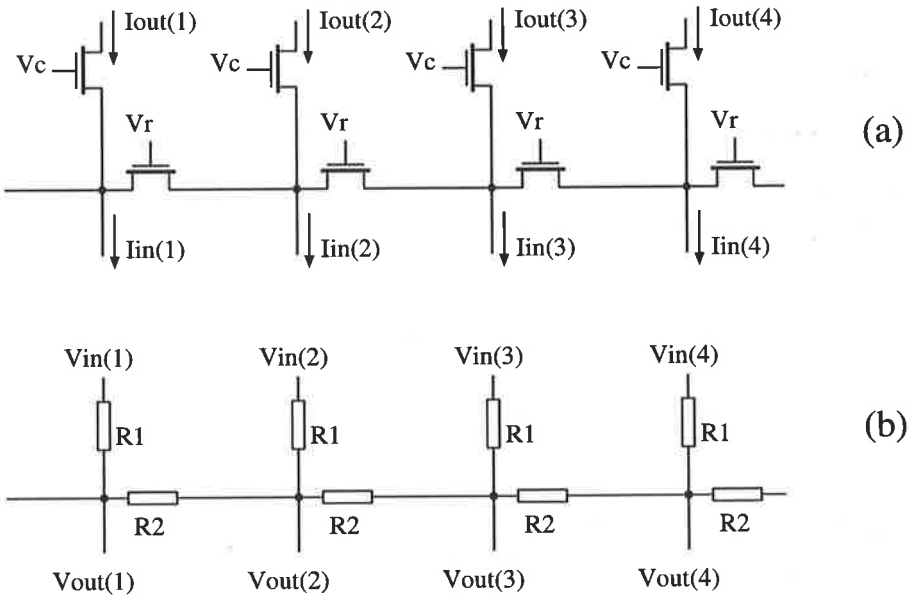


Figure 3.8: a) Current-mode spatial smoothing using diffusive elements. b) Voltage-mode spatial smoothing using resistive elements.

3.5.2 Charge Mode

Representing signals using electrical charge has long been used in CCD devices, although it is not necessarily limited to CCDs. In CCD devices, potential wells hold the charge packets. In standard CMOS or bipolar processes capacitors are used to store and distribute the charge.

Charge mode circuits naturally operate in a switching or clocked mode. In CCD technology charges are transferred and distributed by applying clock signals to the gate of CCD elements. In switched-capacitor techniques charge is transferred by applying a clock signals to switching transistors, which change the topology of the circuit.

From a practical point of view, capacitors are devices with good matching properties, mainly because the matching largely depends on the geometries. Therefore, for applications where matching is important, the switched capacitor approach may be used [Ni et al. 96, Umminger and Sodini 92]. However, the large area of capacitors reduces the number of cells that can be implemented.

Figure 3.9 illustrates a spatial-smoothing function in a CCD and a CMOS implementation. In both implementations, the order of the clock pulses are important. The amount of smoothing can be controlled by the clocking scheme, or applying the clock signals for several more cycles.

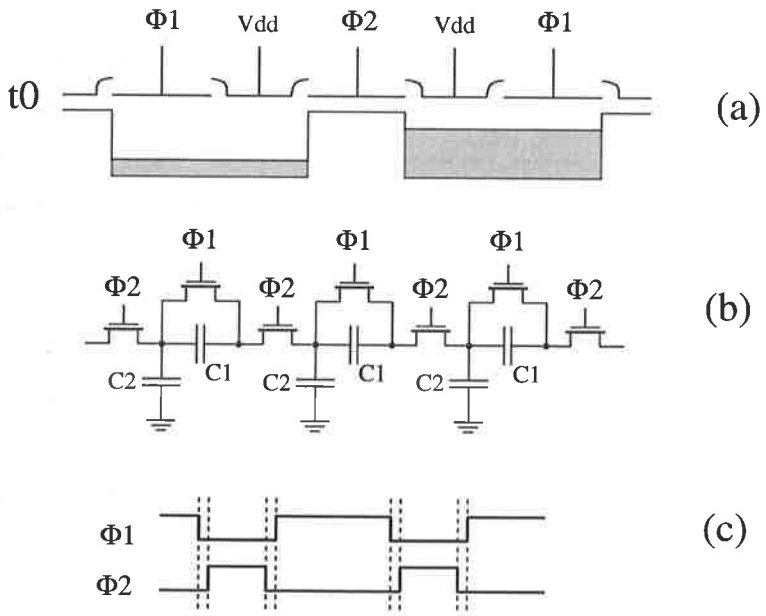


Figure 3.9: Spatial smoothing using a) CCD, b) switched capacitors. c) Clock signals used in the operation.

3.5.3 Mixed Modes

Mixed Analog/Digital

Digital signals are robust against noise and mismatch. Also there are more practical techniques for storing digital signals than analog ones. Digital signals have an inherent advantage when interfacing to host computers. However, analog implementations present larger dynamic range and lower power consumption in a smaller area as compared with digital implementations. Therefore, a mixed analog/digital implementation may use the advantages of both worlds. For example, the weights in a CNN can be applied in a digital format [Dominguez-Castro et al. 97], which may be easily downloaded from a host computer without the need to DACs.

Another approach may consist of using a pulse mode signal representation, which apart from being biologically plausible, is the most efficient way (in terms of interconnection) of transferring analog signals on a single digital line [Murray and Tarassenko 94].

Circuits for converting a digital (or pulse mode) signal to an analog signal are required, when using a mixed analog/digital approach. The limited area in vision chips restricts the resolution of the digital approach to about five bits. In the pulse mode approach the resolution is limited by the minimum and maximum recoverable pulse-width or pulse-frequency which is equivalent to about eight bits.

Mixed Continuous/Sampled

Current mode and voltage mode circuits can be operated either in a continuous mode or a sampled mode. Charge mode circuits are by nature sampled. There is no clear advantage in using sampled circuits over continuous circuits. In switched current (SI) circuits using dynamic current mirrors, the advantage is the reduction of the mismatch in current mirrors because only one transistor is used to replicate the current.

In vision chips it is preferable to minimize the amount of switching, in order to reduce digital noise, and also to reduce the power needed to clock the circuits. If the photocurrent is transduced to a voltage using the integration photocircuit (see Section 5.2.6), then sampling mode can be applied throughout the chip.

In any case, sampled circuits require *hold* elements, which are principally capacitors. This again raises the issue of the area necessary for the capacitors, the switching transistors, and the clock signals.

A Mixture of All

Each of the techniques presented in the previous sections offers certain advantages. Therefore, in a particular design, a mixture of several techniques may be used. Given present technological capabilities, continuous current and voltage mode circuits are advantageous over other

techniques, mainly because they consume the least area. However, as the technology scales down, mixing in other techniques may become feasible.

As an example, (Figure 3.10) spatial smoothing is performed using current mode circuit, whose outputs are integrated, sampled and then further processed in a switched-capacitor network, and finally converted to pulse mode signals and transmitted off-chip using an *address event representation* (AER [Lazzaro and Wawrzynek 95]) method.

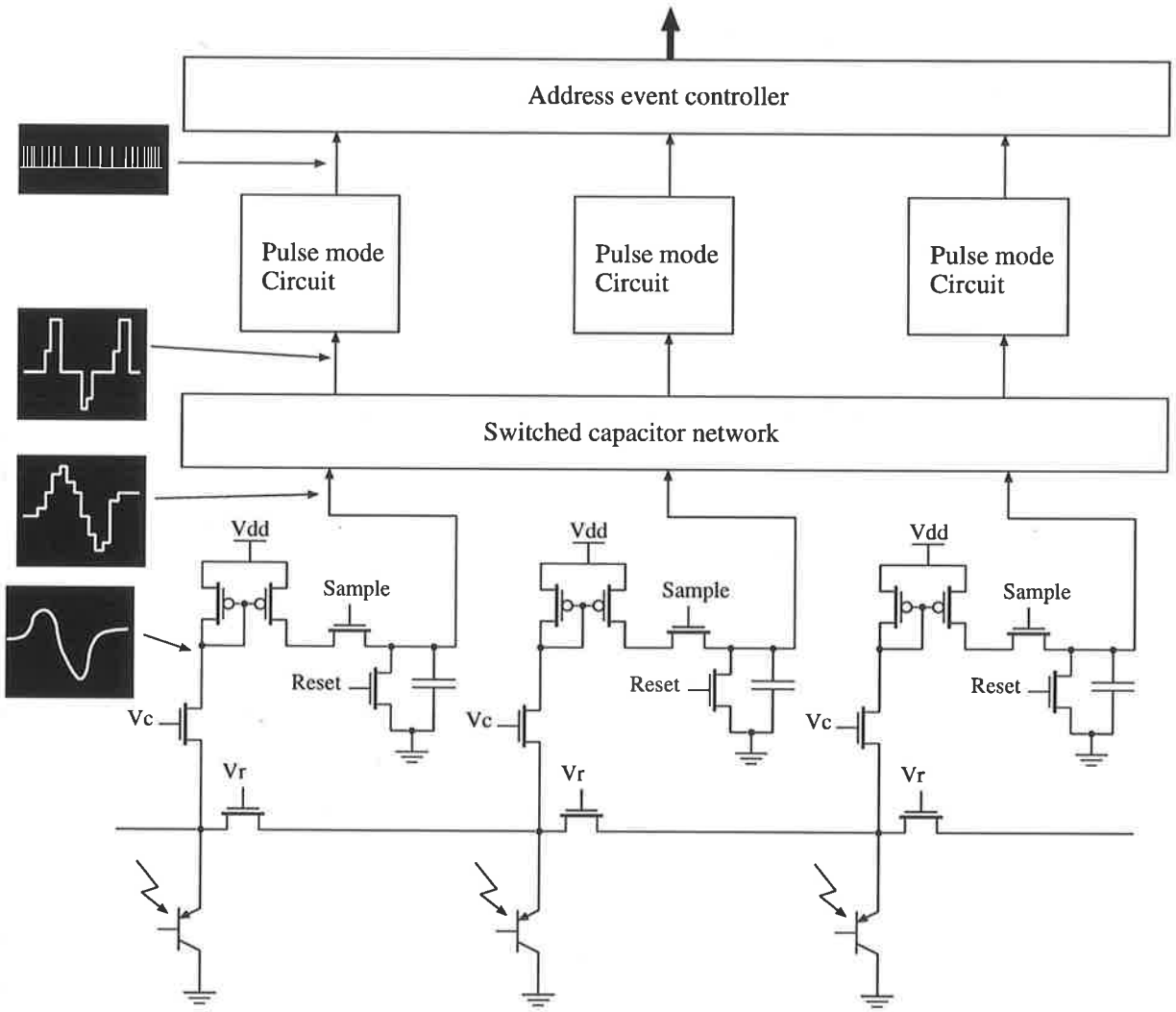


Figure 3.10: An example of a multi-mode approach in the design of vision chips.

3.6 Summary

Table 3.4 highlights the characteristics of the major fabrication technologies used for designing vision chips, and summarizes the points discussed in Section 3.4.

Table 3.5 shows the characteristics of the dominant circuit design techniques used in the design of vision chips. It summarizes the discussions in Section 3.5.

Table 3.4: Process technologies for vision chips.

Process	Features
CMOS	• Mature analog and digital design techniques • Rich set of devices (standard and parasitic), and device equations • Various devices available for analog storage • Standard and low cost process • Low power in subthreshold • Large mismatch in subthreshold
BiCMOS	• All the characteristics of CMOS • Less mismatch if bipolar devices are used • More complex and expensive fabrication process than CMOS
CCD/CCD-CMOS	• Non-standard process (CCD-CMOS) • Limited to charge mode sampling operation • Access to a limited number of devices (CCD) • More complex and expensive fabrication process than CMOS
GaAs	• High speed operation • Capable of emitting light (in specialized processes) • Poor digital and analog performance (in terms of functional reliability) • Poor performance for analog storage • Limited availability of process • More expensive fabrication process CMOS

Table 3.5: Circuit design techniques for vision chips.

Design Technique	Features
Current & Voltage mode	• Large dynamic range (current mode) • Natural form of addition and subtraction (current mode) • Rich set of circuits • No switching noise or clocks required • Lower area consumption
Charge mode	• Less mismatch • Operation controlled by clock signals • Limited number of operations can be performed on charge packets • Switching noise • Higher power dissipation, to drive clock signals • Larger area for capacitors and switching transistors (SC circuits) • Large number of clock signals required (CCD circuits)
Mixed Modes	• **Analog/Digital:** Using the advantages of digital (robustness against noise, ease of storage, and interfacing), and analog (small area, less power, natural form of processing) • **Multi-mode**: Using the advantage of each design technique as suitable • More suitable for scaled processes

Part II

Synthesis of Vision Algorithms

Introduction

So far the design of vision chips has not been formulated into a systematic methodology. This can be attributed to the immaturity of vision chips[1], and to the large variety of vision algorithms, architectures, and circuits that have been used for the implementation of vision chips.

The design flow of a vision chip often follows three major steps as depicted in Figure 3.11.

1. **Algorithm development:** Designing a successful vision chip in the first instance requires a working algorithm. Whether the algorithm has been inspired by biological vision systems or is purely based on mathematical foundations and *assumptions*, its functionality should be shown at least through simulations. The algorithm can be tested and compared against other existing algorithms using some standard benchmarks. Of course, a large body of computer vision is still faced with the lack of test criteria. A digital implementation of the algorithm using a camera and some general purpose or dedicated digital hardware, may demonstrate the functionality of the algorithm in the same environment and situation as the final vision chip will operate.

2. **Hardware implementation:** At this step, the algorithm is mapped into a hardware architecture and consequently into building blocks and low-level circuits. This mapping process may be decomposed into two smaller steps: architectural mapping, and building block mapping. Using certain building blocks may be efficient only for certain architectures and algorithms, and vice versa. Often a demonstration prototype (which could be a small chip) may be useful in evaluating the architecture and the building blocks.

3. **Testing the system:** Testing vision chips is the final step in the design cycle. Testing can be done by applying synthetic or natural images to characterize the performance of the chip. The results can be fed back to the algorithm development and hardware implementation steps to tune the parameters of the algorithm or hardware. Because they have specific image capture and preprocessing circuitry, vision chips may have different spatial and temporal image characteristics than a normal "camera-processor" combination.

[1]Here the term "vision chip" refers to the VLSI implementation of vision algorithms, regardless of it being analog or digital, silicon or GaAs, ...

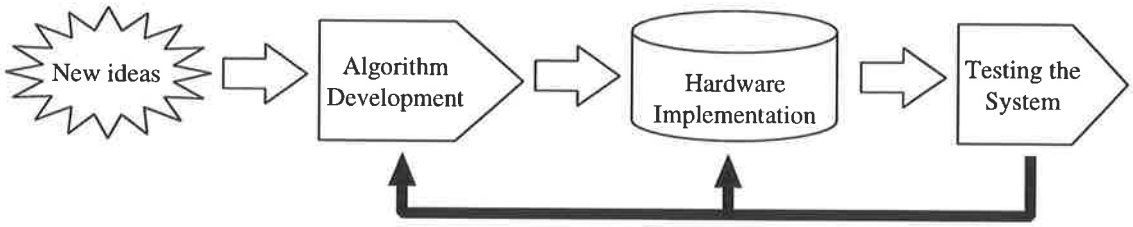


Figure 3.11: Design flow for vision chips.

Implementing a specific computer vision algorithm into VLSI hardware[2] is a difficult step in the development of an artificial vision system. There are many trade-offs that one must consider when designing a VLSI vision chip. Some of the trade-offs could even affect the algorithm. For example, circuit noise and mismatch in an analog implementation may seriously deteriorate the performance of the system, if they are not accounted for in the algorithm. Of course, if some of the main limiting trade-offs, such as area and power dissipation, are ignored, it is not hard to imagine that many algorithms could be mapped into analog circuitry. There already exists a large number of analog circuits for performing almost any mathematical functions, such as sine, cosine, hyperbolic tangent, differentiation, integration, and so on (see for example [Seevinck 88] for some circuits based on the translinear principle). However, when the design trade-offs are taken into consideration, the prospect of applying analog VLSI for vision processing narrows down to relatively primitive tasks.

In this part of the thesis the hardware implementation step in the vision chip design process is discussed. The material is presented in a systematic and general way. Instead of detailing the specific circuits that have designed during this project, they are presented in conjunction with circuits designed by other researchers, and design details are deferred to Part III.

[2]Although a purely digital hardware can also be used for designing vision chips, here only analog implementations, and issues pertaining to this analog VLSI design methodology, are discussed.

Chapter 4

Vision Chips Architectures

An important feature of many vision chips is their architectural efficiency, as compared with a conventional "camera-processor" combination. Most computational or biologically-inspired vision chips possess the same structural format as the algorithm embedded in them. This gives them several architectural advantages, which could be beneficial in many applications.

In this chapter, the architectural aspects of vision chips will be presented from a systems perspective. Issues such as pixel level structures, data flow and data read-out structures, data acquisition methodologies, and system level communication into and out of the chip are described.

A vision chip may be designed without any of these issues being fully addressed, when viewed as an independent component. For the chip to be useful for system level integration, however, these points should be addressed, and also it should be tried to adopt and utilize those mechanisms in which vision chips excelled, such as parallel acquisition and processing of the image, and data redundancy reduction.

4.1 Pixel Level Structures and Tessellation

A vision chip, and in general an imager, performs a mapping of the light intensity from a three-dimensional space into a two-dimensional image in the focal plane. Part of this mapping is performed by the optical system used to focus the image onto the imager plane. Here, the optical system is assumed to be perfectly linear and does not introduce non-ideal effects, such as chromatic aberration and spatial distortion.

Each photodetector takes a spatial sample of the image on a rectangular or hexagonal grid. First we discuss spatial sampling from a purely mathematical point of view, and then consider the effect of physical limitations on the size and spacing of photodetectors, and the advantages and disadvantages of two common pixel tessellation structures in the context of VLSI implementations.

4.1.1 Periodic Spatial Sampling and Tessellation

Ideally, each photodetector performs a Dirac delta-function, $\delta(x, y)$, sampling of the input image. The sampling grid can be rectangular, or hexagonal. Due to its simplicity, the rectangular tessellation is the preferred option. However, when an algorithm relies on distance metrics (such as Euclidean, city-block, or chess-board) between neighboring pixels in the image, this tessellation produces different equidistant maps for each metric. Hexagonal tessellation, on the other hand, has a more symmetric structure in terms of neighborhood connectivity and distance maps. A rectangular tessellation only provides two horizontal and vertical directions. The hexagonal tessellation has three main axes with 60° orientations. There are other tessellation structures, such as triangular, but they are more complicated and do not offer significant advantages for vision chips.

The mathematical treatment of two-dimensional image sampling on different grids can be found in the literature (see for example [Dubois 85]). Here, only two of the most commonly used tessellations, i.e. hexagonal and rectangular, are discussed.

Assuming that the density of the detector per unit area is the same for either tessellation (see Figure 4.1), the spacing between detectors relative to the rectangular tessellation can easily be found and expressed as:

$$sh = \frac{\sqrt{2}}{\sqrt[4]{3}} sr \approx 1.075 sr \tag{4.1}$$

where sr, and sh are the spacing between detectors for the rectangular, and hexagonal tessellations, respectively.

The spatial frequency domain representation, referred to as "reciprocal lattice", of each tessellation is shown in Figure 4.2. The shaded areas show the Voronoi cell of a lattice structure. The Voronoi cell is the common area between all the half-planes created by the perpendicular

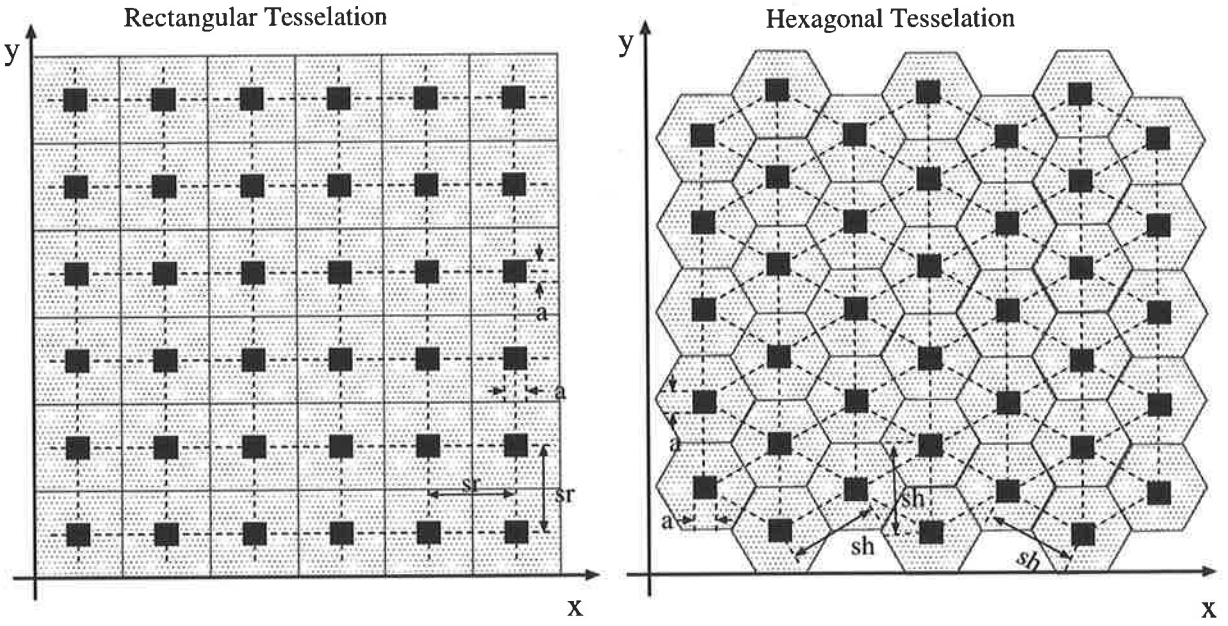


Figure 4.1: Tessellation structures.

bisectors of the lines between the cell and all other cells, which include the cell. The Voronoi cell is in fact the area where the spatial frequency response can extend without any aliasing. The minimum distance of a cell to the edge of its Voronoi cell in the vertical and horizontal directions for the two tessellations, and assuming the same detector density, are:

$$Vr_x = Vr_y = \frac{1}{2sr} = 0.5\frac{1}{sr} \quad \text{for rectangular tessellation}$$
$$Vh_x = \frac{1}{\sqrt{3}sh} = \frac{1}{\sqrt[4]{12}sr} \approx 0.537\frac{1}{sr} \quad \text{for hexagonal tessellation}$$
$$Vh_y = \frac{2}{\sqrt{3}sh} \approx 0.667\frac{1}{sr}$$

Finite Detector Size

The effect of the finite detector size is that of a simple spatial averaging window, which appears as a *sinc* function $(sin(x)/x)$ in the spatial frequency domain. The high frequency tail of the *sinc* function causes aliasing of the high spatial frequency contents of the image, and therefore in theory the detectors should be as large as possible and close to each other. In practice the photodetectors have a finite size and are made fairly large in order to maximize the fill factor, and their shape is usually rectangular or rectilinear. Therefore, in almost all cases there is an intrinsic aliasing due to the finite size of the detector. The aliasing can be reduced only by applying anti-aliasing filters (spatial low-pass filters, for example by blurring the input image).

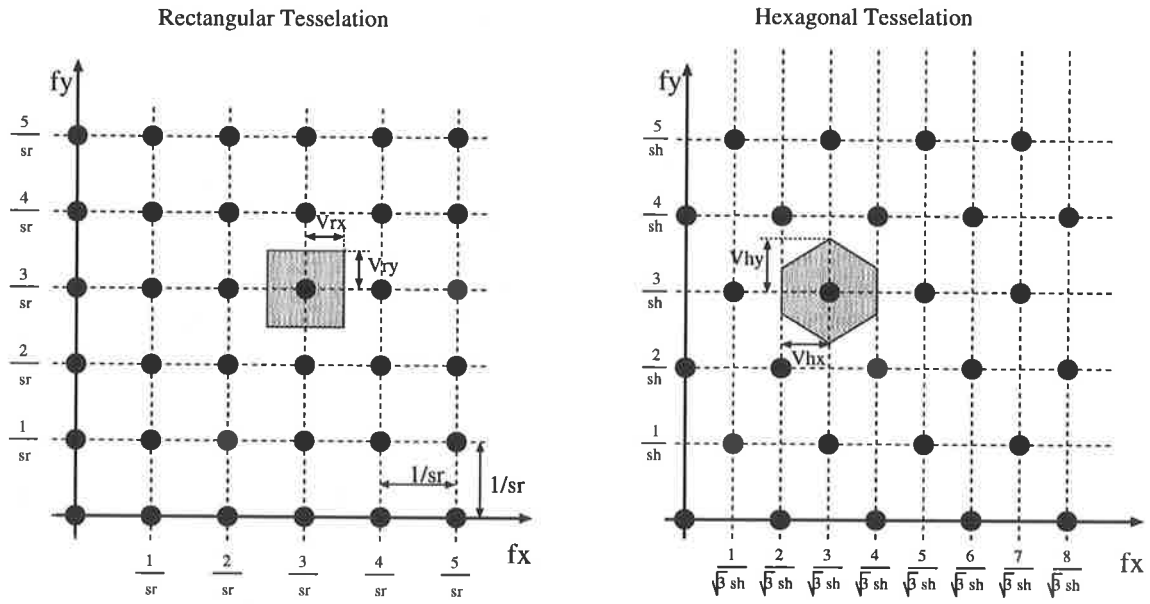


Figure 4.2: Reciprocal lattice for the rectangular and hexagonal tessellations.

4.1.2 Tessellation Structures for Vision Chips

A vision chip may use one of the two common tessellations, i.e. rectangular or hexagonal. We will compare these two structures from different theoretical and practical aspects.

- **Neighborhood distance:**
  Although it may be argued that the neighborhood distances of a hexagonal tessellation are more uniform than the rectangular tessellation, this holds only if all the elements on an equi-radius distance are used. However, if only the elements on the main axes are used in the algorithm, there will be no obvious advantage in using a hexagonal grid, except that a hexagonal grid provides two more pixels than the rectangular grid on each radius. For example, this can result in an increase of $\sqrt{6}/\sqrt{4}$ in the signal to noise ratio if the signals from the nearest neighbors are integrated.

- **Voronoi distance:**
  From equation 4.2 it can be seen that the hexagonal tessellation results in a slightly bigger Voronoi cell than that of the rectangular tessellation in both horizontal and vertical directions. This is usually of concern to imaging technologies, primarily to maximize the spatial frequency content of the image without being affected by aliasing.

- **Layout complexity:**
  Hexagonal structures require interconnections in non-orthogonal orientations with 60 and 30° angles. However, most fabrication processes only allow orthogonal directions for the

alignment of transistors and contacts. If the design requires nearest neighbor interactions, at least six connections will be required for each pixel. If two signals are to be exchanged between each pair of neighbors, twelve connections will be required at the periphery of the cell. Also, for a hexagonal structure three main axes exist, and the power, biasing, and signal distribution along these axes will add to the complexity.

Therefore, in many cases the added layout complexity and the larger number of interconnections required per pixels, makes a hexagonal pixel configuration less attractive than the conventional rectangular configuration.

- **Data manipulation:**
  If further manipulation of the data is required, a simple rectangular tessellation provides a much more convenient representation for the two-dimensional image. Almost all algorithms are formulated in Cartesian coordinates with rectangular grids, which implies that data from a hexagonal sensor would need to be re-mapped.

- **Biological plausibility:**
  Precisely hexagonal or the rectangular structures are not observed in the actual distribution of photoreceptors in the retina of human and other primates. The distribution of the photoreceptor cells in the retina is relatively largely random in all areas [Fatt 78, Eberly et al. 90]. The irregularity has been associated with a minimization of spatial aliasing [Yellott Jr. 83, Yellott Jr. 84], although it has been argued that irregularity always brings a penalty in terms of information theory, and therefore evolutionary processes strive towards regular structures [Bossomaier and Snyder 85]. There are some explanations as to why the photodetectors have been laid out in an irregular fashion [Ahumada Jr. 91].

  The honeycomb structure of the compound eye of insects is tailored to maximize the use of the area of each lens while maintaining maximum symmetry [Horridge 75]. The structure of retinoptic layers in the compound eye has followed the hexagonal structure mainly to comply with the physical distribution, and after a few stages the channels mesh into a microscopically random structure.

  Relatively regular photoreceptor structures, however, are not uncommon in other species other than primates.

To conclude this section, it is clear that unless the slight *theoretical* advantages of the hexagonal tessellation are essential to the performance of the envisaged vision chip, using such a structure is not easily justifiable due to practical limitations in VLSI. Biological systems, as usual, have added to the puzzle of choosing the photoreceptor structure by having different regular and irregular tessellations in the retina of different species.

4.1.3 Spatial Re-Mapping Structures

In conventional imagers and most vision chips, the spatial distribution of the photodetectors forms a regular periodic pattern. A distinct class of imagers perform re-mapping of a two-dimensional image from Cartesian coordinates to other coordinates, such as log-polar and linear-polar. The mapping provided by these sensors facilitates some image processing operations, such as scaling and rotation invariance. Foveated vision chips are a class of space-variant detectors, with a central region of high detector concentration, and lower detector density in the periphery. The most common foveated structures are illustrated in Figure 4.3. The linear-polar sensor in Figure 4.3-a performs the mapping:

$$\begin{aligned} \theta &= \tan^{-1}\left(\frac{y}{x}\right) \\ \rho &= \sqrt{x^2+y^2} \end{aligned} \tag{4.2}$$

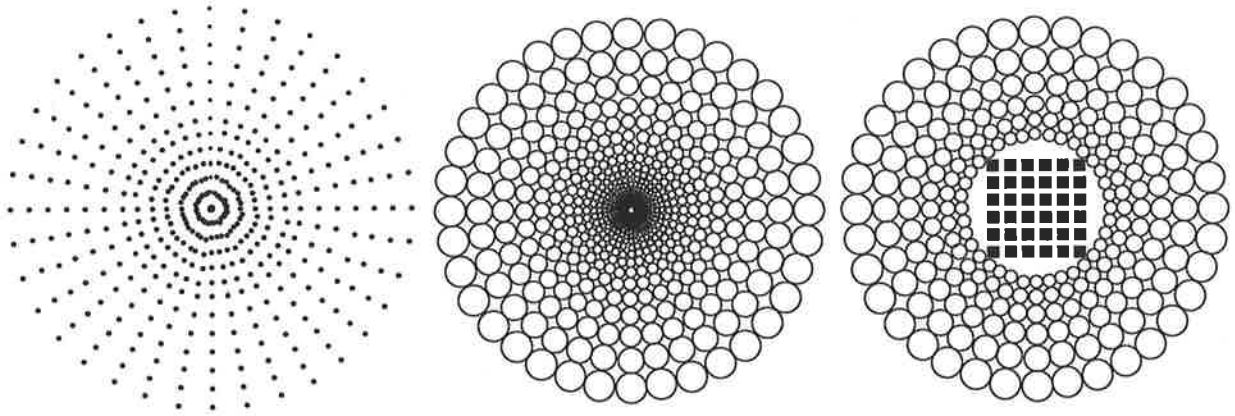
Image rotation in the new coordinates is performed by a simple shift operation in the θ axis. In this sensor all the detectors have the same size. In practical situations, due to the finite size of the detectors, the number of detectors on the central circles is less than those in the outer ones.

The log-polar sensor shown in Figure 4.3-b performs the mapping:

$$\begin{aligned} \theta &= \tan^{-1}\left(\frac{y}{x}\right) \\ \rho &= \log\left(\sqrt{x^2+y^2}\right) \end{aligned} \tag{4.3}$$

In this structure, rotation and scaling is performed by shift operations in the θ and ρ axes, respectively. Again, physical size limits the fine placement of detectors in the center. The third sensor presents a regular and high-density detector structure in the center, and log-polar mapping in the periphery.

These structures are only useful for tasks other than conventional imaging, due to a non-uniform detector distribution. In special image processing applications, such as tracking, the periphery of the detector provides a wide field of sparse detectors. When a target is detected in the periphery, the sensor can be directed toward it so that it is viewed with more detail through the central region. For examples of this type of sensors see [van der Spiegel et al. 89, Wodnicki et al. 97, Pardo et al. 97, Venier et al. 96].



(a) (b) (c)

Figure 4.3: a) linear-polar sensor. b) log-polar sensor. c) log-polar with Cartesian center.

4.2 Sensor-Processor Architectures

The close interaction between photoreceptors and processing elements is the most salient feature of vision chips. Although a fully parallel structure is desirable, in many cases it may not be the optimum solution when considering different trade-offs, such as area, resolution, and power. The best architecture for a specific problem obviously depends on the nature of the algorithm and its spatio-temporal information processing requirements. Therefore, in addition to introducing the most common sensor-processor architectures, different classes of algorithms will also be discussed in this section. Here no distinction is made between analog or digital implementations of the processing elements.

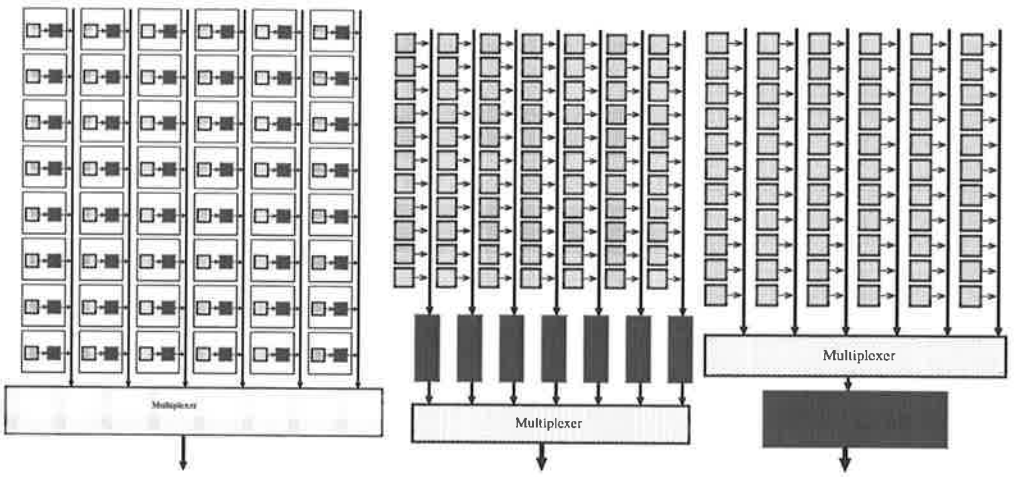
Three levels of sensor-processor interaction can be realized in a vision chip (see Figure 4.4).

1. **Processor per pixel (PPIX):** In this architecture each pixel has its own processing element.

2. **Processor per column (PCOL):** In this architecture there is only one processing element for each column of pixels.

3. **Processor per chip (PCHI):** In this architecture there is only one processor for the entire array.

There exist three general classes of algorithms, based on the usage of spatial and temporal information, and on neighborhood interconnectivity. Feedback has not been explicitly considered here as in terms of VLSI implementations, adding feedback simply means adding extra interconnections between the outputs and inputs of processing elements. These three algorithms are illustrated in Figure 4.5.



(a) (b) (c)

Figure 4.4: Pixel-processor architectures. a) processor per pixel (PPIX), b) processor per column (PCOL), and c) processor per chip (PCHI). The shaded areas represent the processing element(s).

1. **No spatio-temporal processing:** In this class, the output of each pixel at a given time only depends on the value of the input to that pixel at the same time.
$$O(x_i, x_j, t_0) = f_{i,j}(x_i, y_j, t_0)$$

2. **Spatial interactions only without any temporal processing:** In this class, the output of each pixel at a given time depends on the value of the inputs of a neighborhood of pixels.
$$O(x_i, x_j, t_0) = f_{i,j}(x_{i-m}, \ldots, x_{i+m}, y_{j-n}, \ldots, y_{j+m}, t_0)$$

3. **Spatio-temporal processing:** In this class the output of each pixel at a given time depends on the value of the inputs of a neighborhood of pixels within a time interval.
$$O(x_i, x_j, t_0) = f_{i,j}(x_{i-m}, \ldots, x_{i+m}, y_{j-n}, \ldots, y_{j+m}, t_T, \ldots, t_0)$$

4.2.1 Simple Algorithms

The first class of algorithms can be implemented in all three architectures without significant changes. A special architecture can be selected on the basis of the desired silicon area, functionality, and processing speed. For example, if the function to be performed is relatively complicated, it is more economical to use the PCHI architecture. However, if the processing speed of an individual processor creates a bottleneck, then the PCOL and ultimately PPIX architectures should be used. In many cases, the processor can be redesigned for a PCHI architecture in such a way that speed requirements are satisfied.

An interesting and important example of this class of algorithms is the inclusion of A/D conversion on chip. All three architectures have been implemented in several vision chips [Fowler

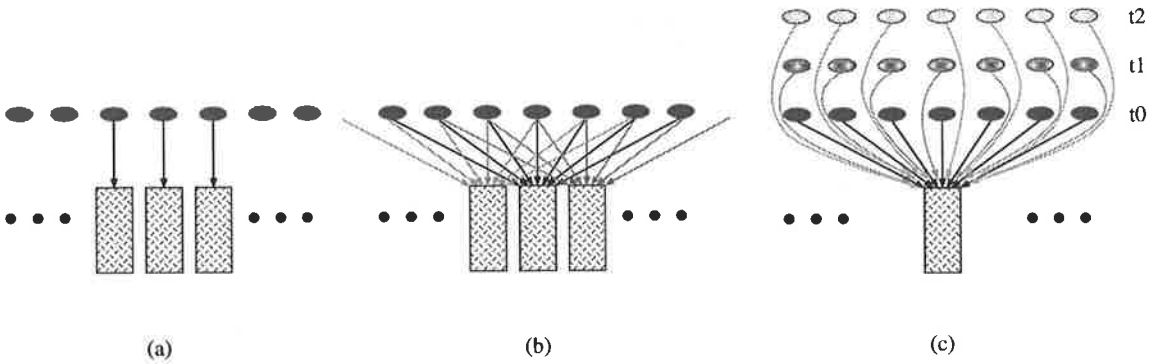


Figure 4.5: Classes of algorithms based on sensor-processor interaction. a) no spatio-temporal interaction, b) spatial interaction only, and c) spatial and temporal interaction. The boxes and ovals represent processing elements and detectors, respectively.

et al. 94, Yang et al. 96, Åström 93]. No significant advantages have been reported for the pixel-level ADC, and the speed advantage of the sigma-delta ADC appears to be balanced by extra processing requirements [Fowler 95]. Torelli points out that there are no significant differences in terms of area, power, and speed, between column-level ADC or single ADC architectures [Torelli et al. 96].

PCHI offers the best solution, when mismatch is a design parameter. The other structures introduce some mismatch.

4.2.2 Spatial Processing Algorithms

The second class of algorithms, in its general form, puts severe demands on VLSI resources. A general spatial processing function in the PPIX architecture requires many interconnections from neighboring pixels. The area of the pixel will be interconnect-limited, unless the algorithm is sufficiently simplified to reduce the neighborhood interactions to the first nearest neighbors. Cellular neural networks (CNN) are an example of this class of algorithms, in which each pixel only interacts with its nearest neighbors [Espejo et al. 94, Dominguez-Castro et al. 97].

A PCOL architecture with processing elements aligned along the X axis has interconnections in the X direction. In order to include information in the Y axis, memory elements should be used to store rows of the image in the neighborhood of the row being processed (see Figure 4.6). By using this approach the PCOL architecture is in fact reshaped to a PPIX architecture along only one column. For the PCHI architecture, memory elements along both X and Y axes are required to store the image information within the required neighborhood.

In both PCOL and PCHI architectures, a serial processing approach can be taken if the same function is to be applied within a neighborhood, apart from scale and offset differences, and

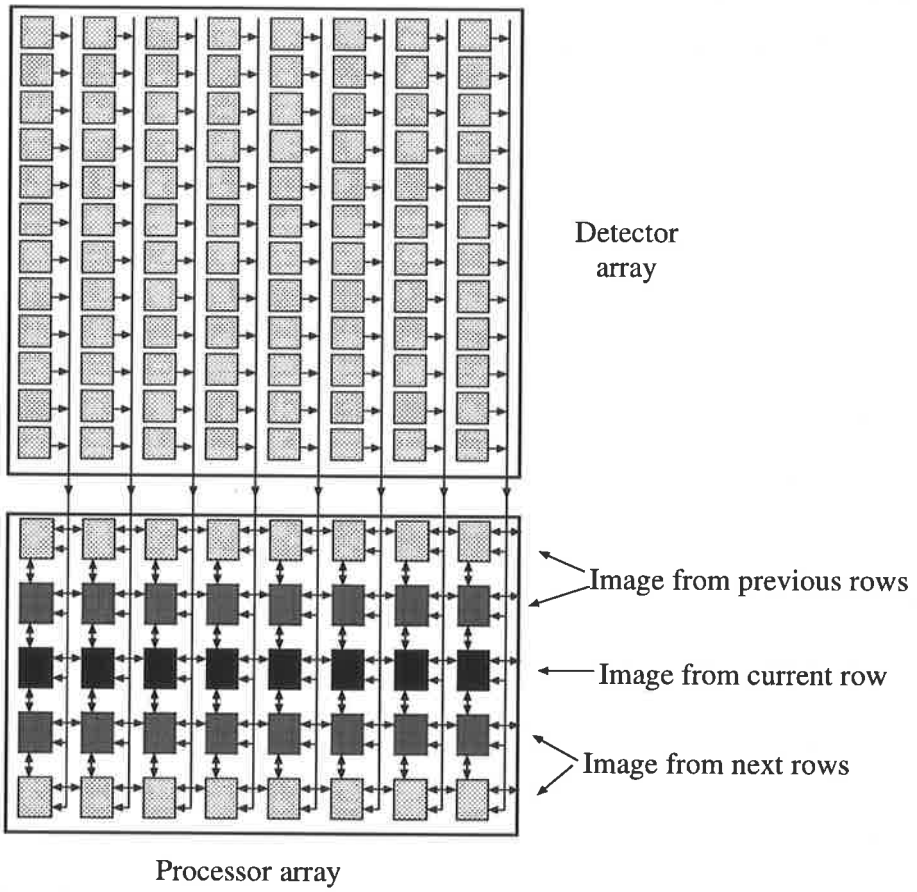


Figure 4.6: Using several rows of processing elements to transform the PCOL architecture for spatial processing along both X and Y axes.

also if the final output is a summation of the output of each pixel.

$$O(x_i, y_j) = \sum_{m,n}[\alpha_{m,n} + \beta_{m,n} f(x_m, y_n)] \tag{4.4}$$

where m and n are in the neighborhood of the i, j pixel. In this case, one can either first store all the inputs and then use a processing element which takes all the inputs in parallel, or apply the inputs one by one to the processor and accumulate the output from all pixels in the neighborhood in a memory element. Although the second approach, which is the approach in serial general purpose computers, requires a smaller processing element, the processing time could be unacceptably long. In fact one of the main thrusts for vision chip development has been the improvement of processing speed by avoiding serial processing. Therefore, this approach would only be suitable if area constraints are very stringent, or processing speed is not a major concern.

4.2.3 Spatio-temporal Processing Algorithms

Including temporal domain information in the processing requires memory elements, whether analog or digital. In a discrete-time framework the number of memory elements required in each cell equals the number of past image frames needed in the processing. For a fully parallel implementation, the number of interconnections per pixel becomes prohibitively large. Therefore, in many real implementations it is only practical to use one past image frame in the processing.

In a continuous-time framework, information is stored in delay elements, which determines the amount of the delay. In general, several delay elements with different time delays are required to convey the information from several time frames.

The comments made in the previous section about spatial processing algorithms will be valid to a large extent because the only difference between spatial and spatio-temporal processing algorithms is the addition of inputs (past image frames).

Memory elements can be implemented either within each cell, or in a separate array. For the simple class of algorithms with only spatial processing in the current image frame, and temporal processing of only the current pixel, i.e.,

$$O(x_i, y_j) = f(x_{(i-m,t_0)}, \ldots, x_{(i+m,t_0)}, y_{(j-n,t_0)}, \ldots, y_{(j+n,t_0)}, x_{(i,t_{-1})}, y_{(j,t_{-1})}, \ldots, x_{(i,t_{-T})}, y_{(j,t_{-T})}) \tag{4.5}$$

either of the two configurations in Figure 4.7 can be used. Two advantages of using a separate array for vision chips are: increased density and fill factor for the photodetector array, and the capability to optimize the memory elements and photocircuits independently [Aizawa et al. 97, Hamamoto et al. 96].

The separate memory array configuration is particularly suitable for the PCOL architecture. However, a combination of PPIX and PCOL architecture can be used for the class described by Equation 4.5.

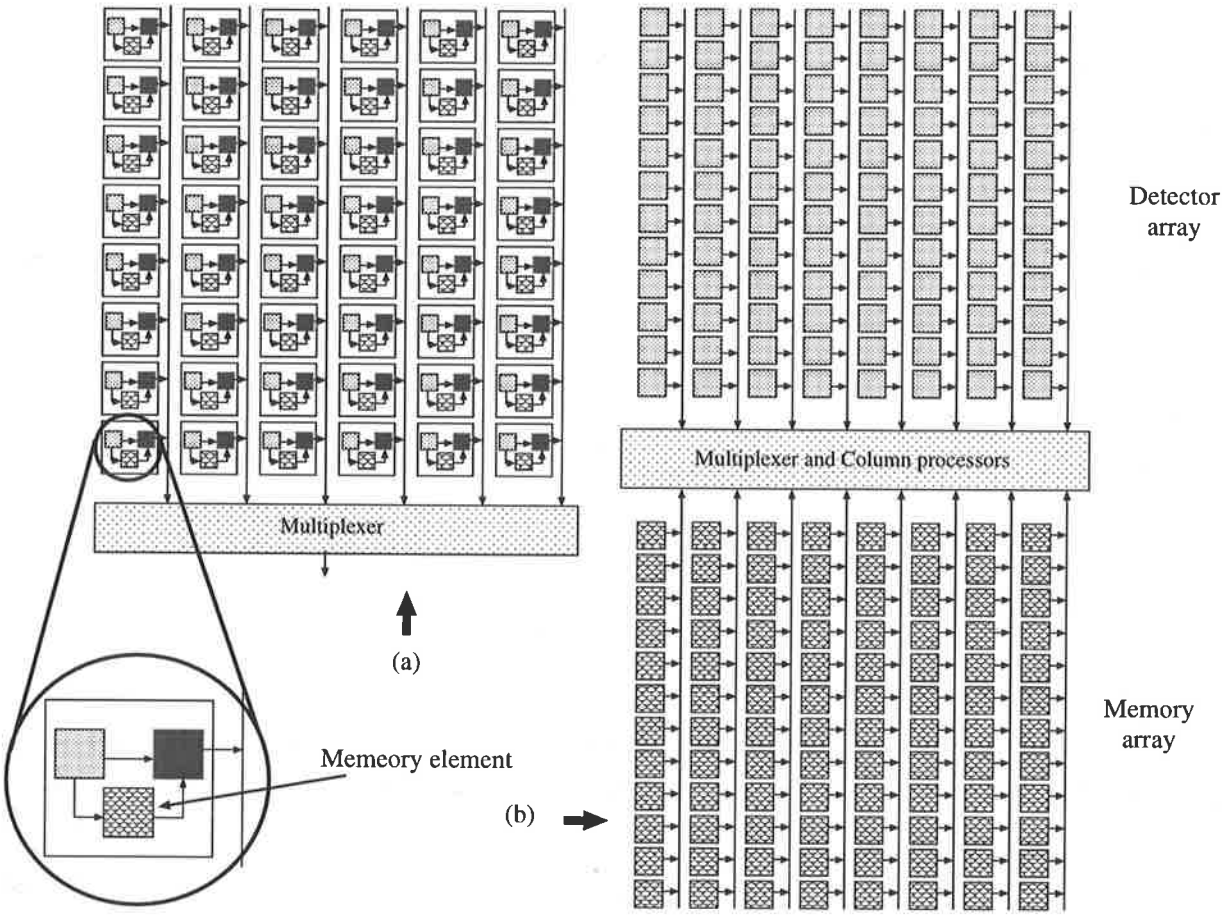


Figure 4.7: Memory element placement for a simple class of spatio-temporal algorithms. a) Memory element inside the pixel, b) Memory elements in a separate array.

In a design a combination of these architectures may be used. For example, a PPIX structure may be used for a simple spatial smoothing operation, a PCOL structure for edge detection along the horizontal axis, and a PCHI structure for analog-to-digital conversion.

4.3 System Level Read-out and Communication

No vision chip is complete without a proper method of delivering its outputs. The method by which the output of a vision chip is read depends on the requirements of the system.

It is necessary to point out that a vision chip is not an imager as in video applications, in which all the pixels should be read out to form a meaningful image. The processing performed by a vision chip is of higher significance, and hence conveying the processed information does not necessarily mean that the output of a vision chip should be read out in the same way as in an imager.

When considering system level communication, two issues are of concern: the interaction between the chip and the system, and the manner in which the chip conveys its output.

4.3.1 Passive & Active Information Transfer

There exist two methods for transferring information between the chip and the rest of the system. In a *passive transfer* mode the chip produces and delivers output at its own pace, and whenever it needs it. For example, the sampling frequency, and the order of sampling of the pixels are set by the chip. In an *active transfer* mode the system directs the acquisition of information from the chip. The obvious advantages of the active method are its more efficient management of information transfer bandwidth transfer energy. Also, for many applications the active transfer method provides a way to reduce the amount of redundant input to the system, thereby increasing processing performance.

The active transfer mode is especially important for vision chips whose output is in a two-dimensional image format, and hence produce a large amount of information. Those chips that only yield global information about the image, such as global intensity, global velocity, or global rotation and position values, produce very little data.

4.3.2 Data Read-Out

The advantage of processing data in parallel may be lost if there does not exist an efficient way of transferring information from the chip to an outside system. For vision chips which produce a two-dimensional output, several methods exist to scan the data out of the array for transferring it off the chip.

1. **Scanning using decoders:**
   In this method each cell is selected by two N-to-2^N decoders in the X and Y axes. In a PCOL architecture (see section 4.2) the decoder in the X direction selects the output of the processing elements. For random access read-out this method is the only choice.

2. **Scanning using shift registers:**
   In this method a "1" is shifted through the shift register, and rows and columns are selected sequentially. Using this method several rows or columns can be selected simultaneously, and if the output of the cells is a current then summation of the output of the selected cells can be realized. An advantage of this method over the decoder-based method is the small number of lines required to control the operation of the shift register.

3. **Synchronous address-event representation (SAER):**
   In a conventional image sensor, every output is read out, even though in many cases only a fraction of the pixels may indicate the occurrence of a spatial or temporal event. If a pixel activates a flag on the occurrence of an event, only those pixels with an activated flag need to be read, and a significant reduction in data acquisition time can be achieved. The values of the activated pixels along with their address are sent off chip. This method has been used for motion detection and compression chips [Aizawa et al. 94].

   However, in SAER each pixel requires some extra circuitry to detect the event and activate the flag signal. The read-out multiplexers should also be able to detect the activated flags and by-pass the reading of the rest of the pixels.

4. **Asynchronous address-event representation (AAER):**
   In SAER events are checked during the read-out phase. In AAER, whenever an event at a pixel occurs the value of that pixel is read out immediately. In a real situation several pixels may simultaneously be triggered by an event, so special arbitration circuits are required to allow only one cell at a time to be read out. The basic arbitration circuit checks the *acknowledge* and *request* signals for a pair of cells and only selects one of them. After the output of the selected cell is read, the other cell is selected. The arbitration circuits form a binary tree, to ensure that only one cell can be selected.

   AAER has been applied to silicon auditory processors [Lazzaro et al. 93], but can also be used for any one or two-dimensional arrays, including vision chips. In a simplified version of this method, the analog signal values are represented using the pulse-mode signals. Therefore, each pulse triggers an event. As the only important information from a pulse in this representation is its location (address) within the array, and its occurrence time, only the address needs to be transmitted over the channel. However, the speed of the AER circuits should be higher than the rate at which the events occur, in order to preserve the temporal information in each pulse.

5. **Multi-channel read-out:**
   In each of the above methods the transfer bandwidth can be increased by using multiple output lines. This is often more economical for the first two methods, as they do not require the address lines to be sent off chip along with each data channel. Multi-channel

read-out is also more economical for chips with analog outputs, as each output requires only one line for single-ended signals, or two lines for differential signals.

A widely demanded feature of vision sensors is the **random-accessibility** of individual pixels. The decoder-based scanning method provides this feature at no extra cost, while the shift register-based method requires a cumbersome control logic to provide full random access. The two other methods, i.e. SAER and AAER, cannot be used in a random access mode, as the location of the pixel which is read out, depends only on its activity.

Windowing on a small region of the array is another feature, which is often used in tracking systems. The decoder-based scanning method intrinsically supports this feature. If the window is scanned serially, the shift-register scanning can be easily modified to support windowing. The address-event methods can support this feature only by looking at those outputs whose address is within the required region. This can be achieved by ignoring the events coming from areas outside the required window.

4.4 Summary

This chapter provided insights into architectural aspects of vision chip design at several levels.

The study of tessellation structures suggests that it is unnecessary to use hexagonal tessellations, unless the few and relatively insignificant advantages of hexagonal tessellation are crucial to the performance of the vision chip.

The PPIX, PCOL, and PCHI architectures, which represent the type of pixel-processor interaction, were discusses. Three classes of algorithms, based on their requirements for spatial and temporal information, were also identified, and the mapping of each class of algorithm to each of the three architectures was discussed. And finally, several techniques for transferring information from vision chips were introduced.

Chapter 5

Building Blocks for Vision Chips

In this chapter building blocks for designing various parts of a vision chip are described. These include photodetectors and photocircuits, circuits and networks for spatial, temporal, and spatio-temporal processing of the input image, and circuits for data read-out.

An essential part of vision chip design, which may easily be overlooked, is circuit complexity. As we go through this chapter we can see that a great amount of effort is spent on designing *small circuits* with maximum functionality, minimum power dissipation, and maximum dynamic range and functional robustness. Among these factors, area remains the most limiting, and the most important trade-off in realizing large vision chips. In clear terms the lavish expenditure of transistors (e.g. as seen in conventional operational amplifier design) cannot be enjoyed at all in designing vision chips. The area of a pixel in a vision chip is quadratically proportional to the number of transistors used in its implementation [Boahen 96].

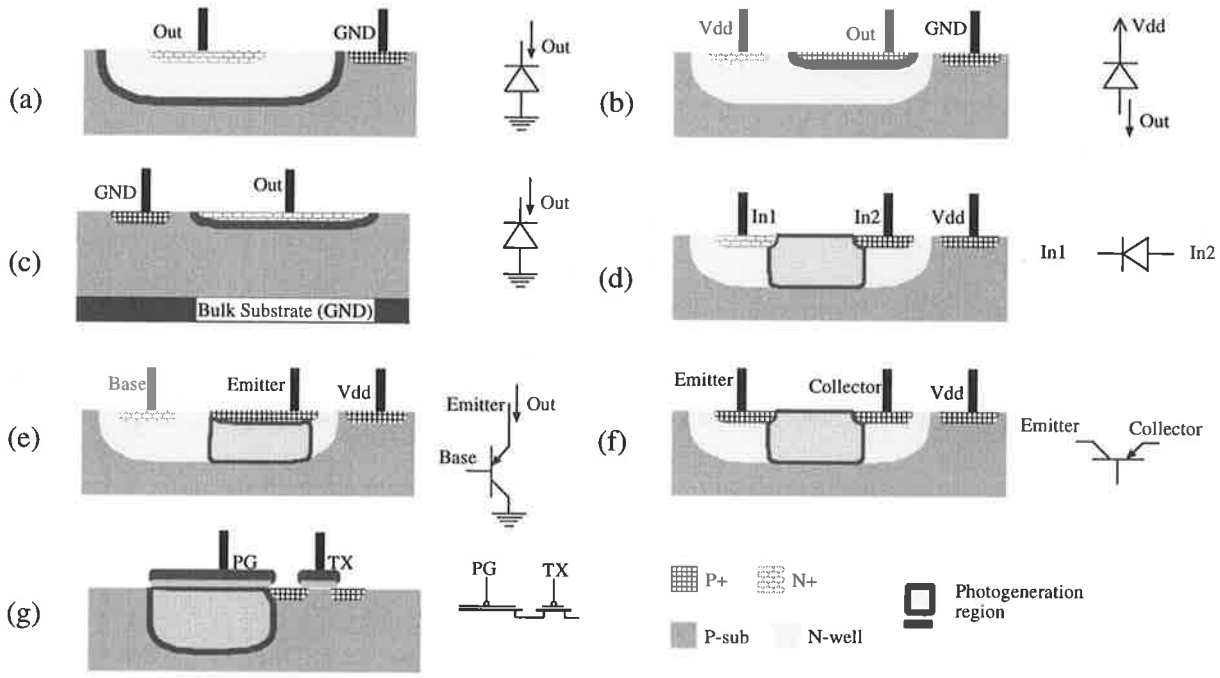


Figure 5.1: Photodetector structures in standard CMOS processes. a) well-substrate diode, b) diffusion-well diode, c) diffusion-substrate diode, d) lateral diode, e) vertical bipolar transistor, f) lateral bipolar transistor, and g) photogate.

5.1 Photodetectors

Photodetectors are the doorway to vision chips. Any imperfection at this stage, with respect to desired characteristics, cannot be compensated, even with *a priori* knowledge, or may be approximately compensated, to some extent, but at a high computational cost. The characteristics of the detectors, such as bandwidth, noise, linearity, and dynamic range, directly affect the performance of the system. Therefore, it is highly desirable to have as perfect a photodetector as possible.

I will only discuss the photodetector elements found in CMOS, as the majority of vision chips are implemented in CMOS. In a CMOS process, there are at least three vertical and one lateral junction diodes, one lateral and one vertical bipolar transistor, and a photogate structure, which can be used as photodetecting elements (see Figure 5.1). Junction devices are often used in the photocurrent mode. The photogate is operated in a charge integration mode, where initially the potential well under the gate is emptied of charges and then a large voltage is applied to the gate to create a potential well to collect the charges. After the integration cycle the charges are read out.

A detailed analytical treatment of these structures is presented in Appendix A, mainly to provide an insight into the performance of these devices. Here, only present the simulation

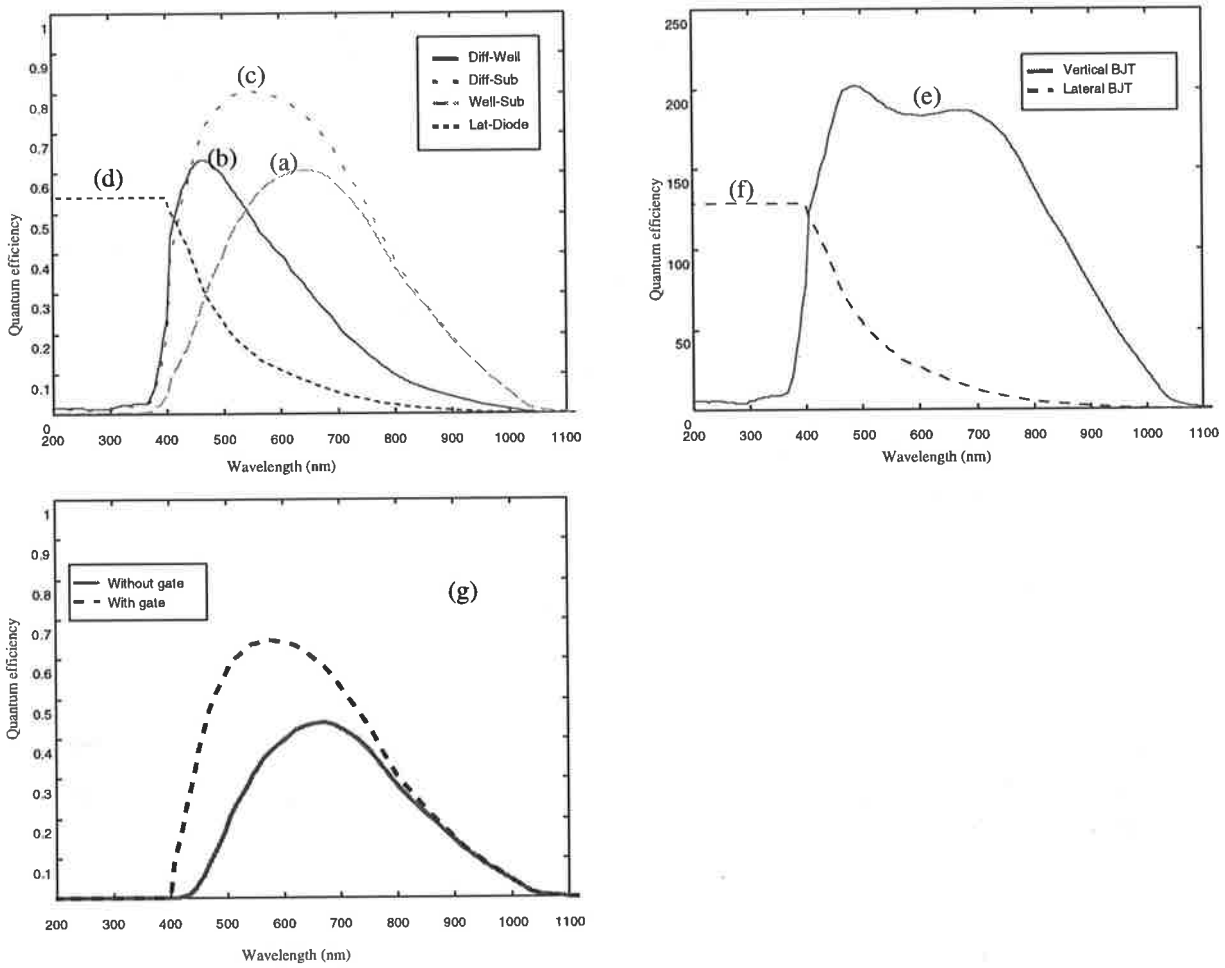


Figure 5.2: The simulated quantum efficiency of the devices shown in Figure 5.1.

results are presented, and some of the important characteristics of each detector in terms of quantum efficiency and dark current, are summarized.

The simulated quantum efficiency (QE) of each of the illustrated devices in a standard 2μm p-well CMOS process is shown in Figure 5.2. The following conclusions can be made based on the simulation results.

- Among the vertical diode structures, Diff-Sub diode has higher QE than the other structures. The Diff-Well diode has a better response in the blue region than the Well-Sub diode because the junction is closer to the surface and therefore can absorb more short-wavelength photons, which have a high absorption coefficient.

- Both the vertical and lateral bipolar transistors have a higher-than-one QE as a result of the current gain of the transistors.

- The lateral diode and lateral BJT both have good blue response, as the electron-hole pairs generated by short-wavelength photons can contribute to the photocurrent. The QE for

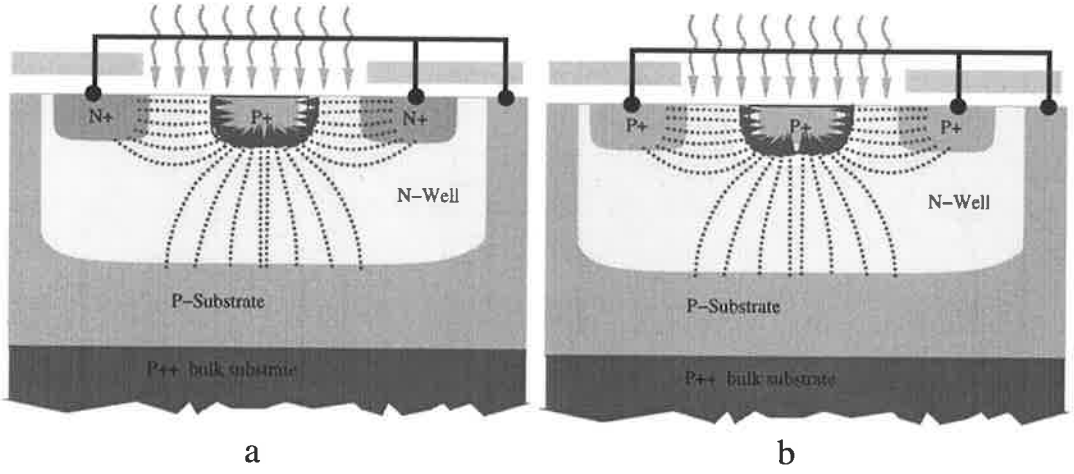


Figure 5.3: Mixed photodetector structures showing the contributions from the lateral and vertical junctions. a) a mixed junction diode, and b) a mixed BJT.

these devices falls exponentially at wavelengths larger than 400 nm.

In the simulations of the lateral structures it was assumed that only the region between the two diffusions was exposed to light, otherwise the effect of vertical junctions should also have been considered. The blue response can be improved, by mixing the vertical and lateral junction devices, as shown in Figure 5.3.

- The photogate has the smallest QE because the photocurrent filling the potential well is mainly from the diffusion of carriers, and also the gate material blocks most of the short and medium-wavelength photons. In silicided processes, the absorption of the silicide is so high that it should either be removed, or several windows be made to allow light to reach the semiconductor.

Table 5.1 presents the simulated dark currents for each device. Among the vertical diodes the current of the well-substrate is the smallest, due to the lower doping densities in those regions.

The bipolar transistors have a dark current about two orders of magnitude larger than the photodiodes, which severely limits the performance of integration based photocircuits (see Section 5.2). A large dark current can more quickly charge the input capacitance in integration based photocircuits, and hence, saturating the input node will require a shorter integration time than for photodiode structures. However, in photocircuits which operate in continuous current-mode, the higher input photocurrent would prevent the circuits from operating in the deep-subthreshold region.

The small dark current of the photogate structure is due to the fact that there is no junction present in the device and the dark current is due only to the carrier recombination in the substrate.

Table 5.1: The simulated dark current for CMOS photodetector devices in a 2μm CMOS process.

Well Substrate	$2.7 \times 10^{-18} A/\mu m^2$
Diffusion Well	$4.8 \times 10^{-18} A/\mu m^2$
Diffusion Substrate	$2.6 \times 10^{-17} A/\mu m^2$
Lateral Diode	$7.0 \times 10^{-22} A/\mu m$
Vertical BJT	$5.3 \times 10^{-16} A/\mu m^2$
Lateral BJT	$8.3 \times 10^{-20} A/\mu m$
Photogate	$2.9 \times 10^{-19} A/\mu m^2$

5.2 Photocircuits

Photocircuits are the front-end circuits that receive and preprocess the photocurrent. The way in which the input photocurrent is processed depends on the overall architecture of the vision chip. For example, in spatial vision chips the DC level of the inputs are important, and should therefore be preserved by the photocircuits. For spatio-temporal processing vision chips, the photocircuit should preserve the temporal characteristics of the input as well.

One of the most important aspects in the design of photocircuits is the dynamic range. Photocircuits should have a large operating dynamic range and be able to provide a compressed output so that further processing circuits do not need to cope with a large dynamic range input. They should also provide necessary amplification and buffering, if required. These requirements should all be satisfied while considering the area limitation of the photocircuit.

In the following sections some of the photocircuits which are widely used in vision chips are described.

5.2.1 Logarithmic Compression Photocircuit

Figure 5.4 shows the logarithmic compression photocircuit. If the input photocurrent is very low so that the transistors operate in the subthreshold region, the output voltage will be logarithmically dependent on the input current. Several MOS diodes in series can be used to increase the sensitivity of the circuit. For the circuit in Figure 5.4-a, and assuming that the transistors

operate in the subthreshold region we will have

$$I = \begin{cases} \frac{W}{L} I_{D0} e^{\frac{V}{U_T}\frac{1}{n}} & \text{(one MOS diode)} \\ \frac{W}{L} I_{D0} e^{\frac{V}{U_T}\frac{1}{n^2+n}} & \text{(two MOS diodes)} \\ \frac{W}{L} I_{D0} e^{\frac{V}{U_T}\frac{1}{n^3+n^2+n}} & \text{(three MOS diodes)} \end{cases} \tag{5.1}$$

where W and L are the width and length of the transistor, respectively, I is the input photocurrent, n is the subthreshold slope factor, I_{D0} is a process dependent parameter, and V is the output voltage. The small signal response of both photocircuits can also be derived as

$$\frac{v_o(s)}{i_i(s)} = \frac{-1}{g_m + C_p s} \tag{5.2}$$

where C_p is the parasitic capacitance at the input node, and g_m is the transconductance of the transistor. Note that in subthreshold region g_m is directly proportional to the current level. Therefore, at low light levels the circuit will have a slow response.

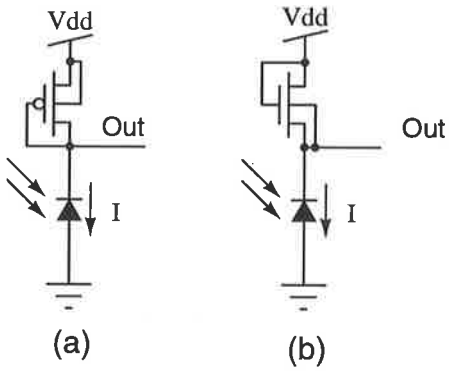


Figure 5.4: Logarithmic compression photocircuit. a) Using a P-MOSFET, and b) using an N-MOSFET.

5.2.2 Logarithmic Compression With Feedback Amplifier

Both circuits in Figure 5.4 can be modified to enhance their small signal characteristics, by using the amplifiers in a feedback loop, as shown in Figure 5.5. The small signal response of these circuits are:

$$\frac{v_o(s)}{i_i(s)} = \frac{K}{[C_p + (K+1)C_{gs}]\left[s + \frac{g_m K}{C_p + (K+1)C_{gs}}\right]} \approx \frac{1}{C_{gs}\left[s + \frac{g_m}{C_{gs}}\right]} \quad \text{For Figure 5.5-a}$$

$$\frac{v_o(s)}{i_i(s)} = \frac{-K}{[C_p - (K-1)C_{gd}]\left[s + \frac{K g_m}{[C_p - (K-1)C_{gd}]}\right]} \quad \text{For Figure 5.5-b} \tag{5.3}$$

The transfer function of the circuit in Figure 5.5-a indicates a clear improvement, as the pole of the circuit has now moved from $-g_m/C_p$ to $-g_m/C_{gs}$ (Note that $C_p \gg C_{gs}$). The location of the pole still depends on g_m, which is in turn dependent on the input photocurrent level.

The circuit in Figure 5.5-b will be unstable if the gain K of the amplifier is too large. The circuit is stable if $K < \frac{C_p}{C_{gd}} + 1 \approx \frac{C_p}{C_{gd}}$.

The voltage at the input node in both circuits is almost fixed at V_{ref}. The DC response of the circuits is the same as that of the logarithmic compression circuit.

The amplifier in Figure 5.5-a can be a CMOS inverter, or a simple two-transistor amplifier. In both cases, the DC level at the input node, V_i, will be set at the transition voltage of the inverter.

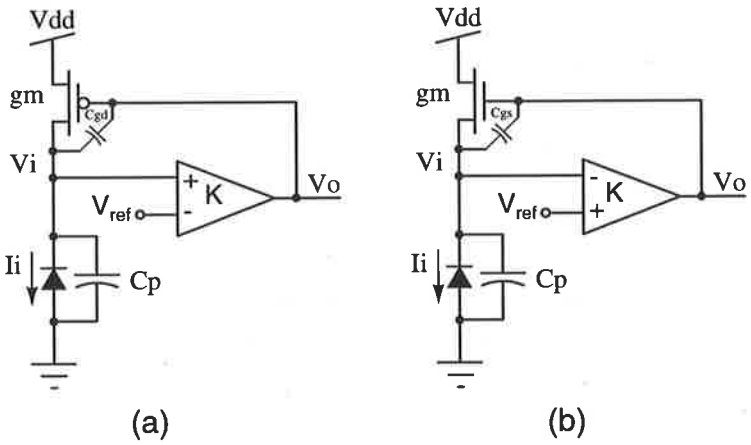


Figure 5.5: Logarithmic compression photocircuits using a feedback amplifier.

5.2.3 Buffered Logarithmic Photocircuit

In order to drive large loads, a buffered photocircuit is required to provide a low output impedance. The circuits shown in Figure 5.6 can be used for this purpose. Circuit 5.6-a is the logarithmic compression photocircuit followed by a buffer, while circuit 5.6-b is a special case of circuit 5.5-b, when $K \approx 1$.

The advantage of using circuit 5.6-b is that, the output voltage does not observe the voltage drop caused by the buffer, and does not get affected by the nonlinearities in the buffer circuit.

Circuit 5.6-a has a voltage drop of approximately V_T, and therefore more than one MOS diode in the input is required, otherwise the output voltage swing may reach the supply voltage. The photocircuit in Figure 5.6-c alleviates this problem because the direction of the voltage drop is opposite to that of the buffer in Figure 5.6-a. However, as this photocircuit uses transistors of different types, the layout design rules will not allow a compact layout. To use transistors of the same type, the logarithmic photocircuit of Figure 5.4-b must be used instead.

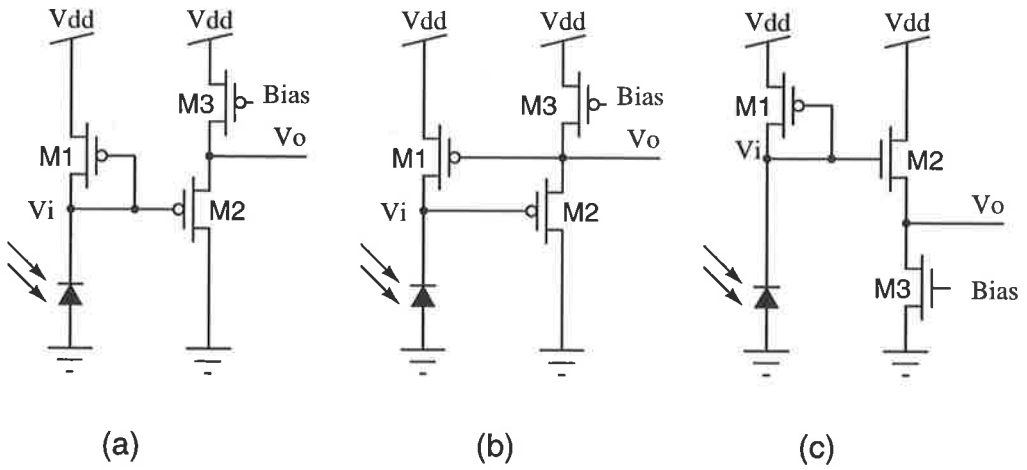


Figure 5.6: Buffered photocircuits. a) using PMOS buffer. b) using a feedback buffer. c) using NMOS buffer.

It is trivial to show that the AC response of the circuits without any load is very similar to that of the simple logarithmic compression photocircuit.

Moreover, circuit 5.6-b shows a clear advantage, when mismatches are considered. The mismatch at the output voltage only depends on the mismatch of the transistor M1, while other circuits will be affected by the mismatch in the buffer circuit as well. Monte carlo simulations were performed on both circuits. They indicate about four times more mismatch in other photocircuits compared with the circuit 5.6-b.

A clear conclusion of this section is that circuit 5.6-b is superior to all other buffered photocircuits in many respects.

5.2.4 Adaptive Logarithmic Photocircuits

A disadvantage of all the logarithmic compression circuits is the extreme compression, as the output voltage of the photocircuit with one diode only varies by one volt for an input photocurrent range of over six orders of magnitude. Hence, further spatial or temporal processing circuits will need to be very sensitive, in order to resolve temporal contrast. Therefore, it is desirable to produce an output with high temporal contrast (amplified) while still maintaining a certain level of compression of the intensity level.

This can be achieved by introducing an adaptive element to the feedback path of the circuits described in section 5.2.2. The simplest adaptive element is shown in Figure 5.7-a. If the resistor R is very large then the transfer function of the circuit will depend on the capacitive ratio $\frac{C_1}{C_2} \ll 1$. If used in a feedback path, this element will create an AC gain of about $\frac{C_2}{C_1}$, while the DC operating point will be unaffected by the adaptive element, provided that its output does not observe any DC load.

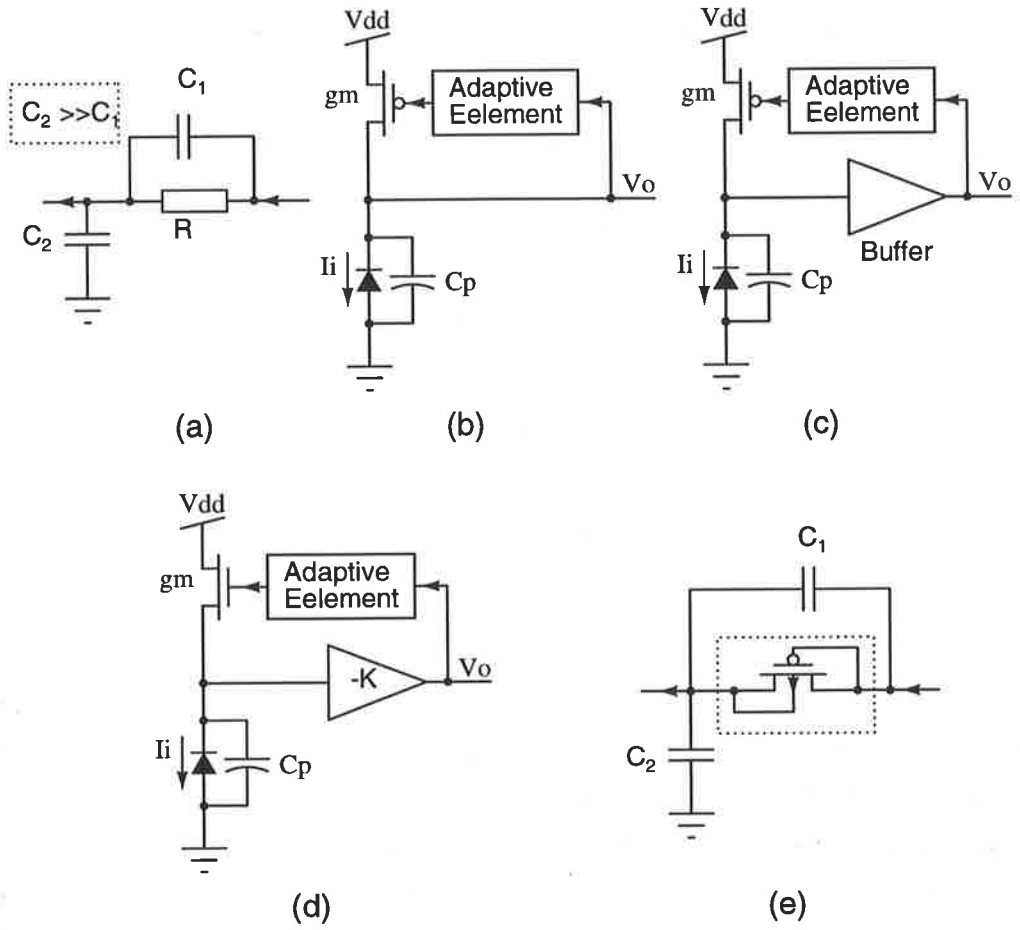


Figure 5.7: a) an adaptive element. b) including the adaptive element in a simple logarithmic compression photocircuit. c) using the adaptive element in a buffered photocircuit. d) using the adaptive element in an feedback-amplifier photocircuit. e) an implementable adaptive element.

Using the adaptive element in Figure 5.7-a, we get

$$\frac{v_o(s)}{i_i(s)} = \frac{-R_o}{1+\dfrac{g_m R_o}{1+\dfrac{RC_2 s}{1+RC_1 s}}} \approx \frac{-R_o}{1+R_o g_m \dfrac{C_1}{C_2}} \quad \text{For circuits 5.7-b and c}$$

$$\frac{v_o(s)}{i_i(s)} = \frac{KR_o}{1+\dfrac{g_m R_o K}{1+\dfrac{RC_2 s}{1+RC_1 s}}+R_o g_m} \approx \frac{C_2}{g_m C_1} \quad \text{For circuit 5.7-d } R \gg 1 \quad C_2 \gg C_1 \tag{5.4}$$

where R_o is the impedance seen at the drain of the transistor including the parasitic capacitance and the output impedance of the transistor.

A large resistance is required to satisfy the time constants required for the adaptation purpose (in the order of several milliseconds to several minutes). However, this is not readily

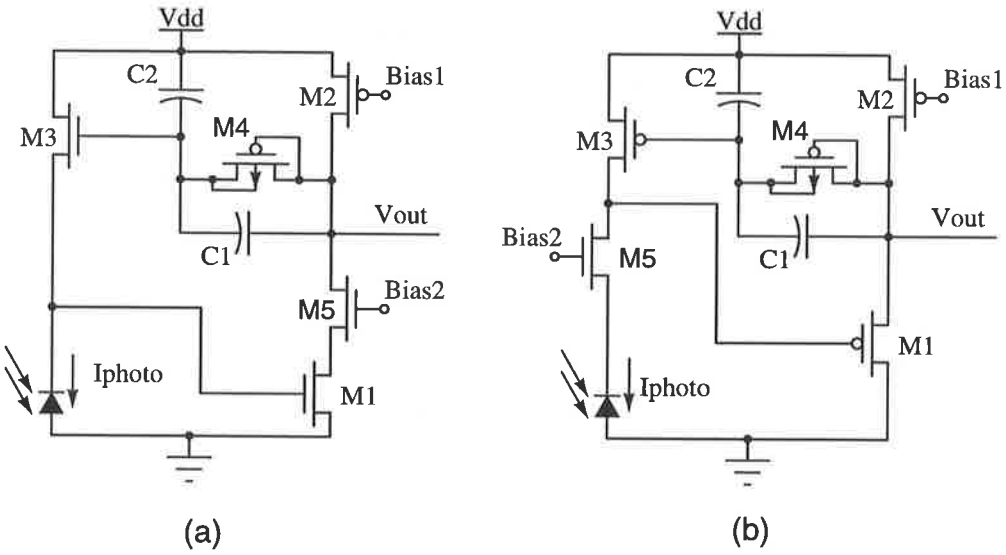


Figure 5.8: a) Delbrück's adaptive photocircuit. b) An adaptive buffered photocircuit.

implementable using conventional circuits. Of the several devices used by Tobi Delbrück to implement the adaptive element [Delbrück 93], the element in Figure 5.7-e appears to be the most suitable one. The transistor in this element has an *expansive* characteristic in that the current-voltage relationship is exponential. This *expansive* characteristic enables sharp transients to be dampened very quickly, while making the settling time to the ultimate DC level very long (this settling time depends on the leakage current of the device which is typically in the order of femto amperes).

Two adaptive photocircuits are shown in Figure 5.8. Circuit 5.8-a is Delbrück's original photocircuit, and circuit 5.8-b is the realization of the circuit shown in Figure 5.7-c. Transistor M5 is used as a common-gate stage in both circuits, which function almost identically. However, at very low input photocurrents (less than 10pA), the response of circuit 5.8-b is slower. Circuit 5.8-b is less sensitive to mismatch as discussed in section 5.2.3. Mismatch in circuit 5.8-a depends on the matching properties of all transistors in the circuit, while in circuit 5.8-b it only depends mainly on the matching of M3. Simulations indicate more than four times less mismatch in this circuit compared with Delbrück's photocircuit.

5.2.5 Current Amplifier Photocircuit

Light intensity inside a normal room is usually below 1 lux, and at these levels photocurrents are in the deep-subthreshold region (< 1 nA). It is therefore desirable to increase the photocurrent similar to photo-multiplier devices, as shown by the two simple circuits in Figure 5.9. The input current is injected into the base of a bipolar transistor, whose current gain (around 100)

boosts the photocurrent. Note that this will amplify the signal and noise at the same time, and hence the signal to noise ratio remains relatively constant. However, by increasing the signal levels, subsequent circuits will not need to cope with very small currents. By using a BJT as the photodetector, as shown in Figure 5.9-b, the current gain can be boosted further by the current gain of the transistor.

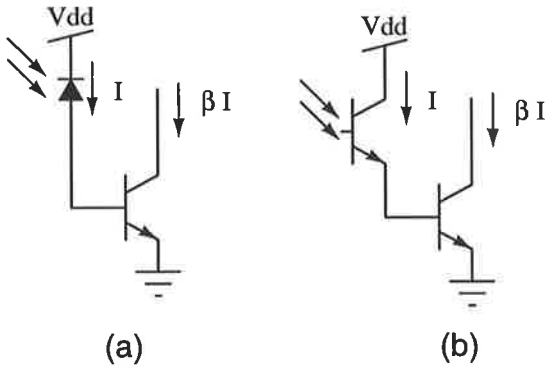


Figure 5.9: Photocurrent amplification. a) using a photodiode, and b) using a photo-BJT.

5.2.6 Charge-Integration Photocircuit

Transducing the charge (or current) into voltage can be achieved by means of a charge integration photocircuit, such as that shown in Figure 5.10. Initially, the reset transistor is on and the voltage at the input node In is set to the reset value. The integration cycle starts when the reset transistor is turned off. The input capacitance usually consists of parasitic capacitances of the devices connected to this node (C_p). In some processes, e.g. GaAs, which have a large gate leakage current, an additional capacitor, C_s may be needed to ensure enough charge storage time.

The advantages of this photocircuit are its linear charge-voltage transfer characteristics (provided that the capacitor is linear), controllable dynamic range by changing the integration time, and low sensitivity to device mismatch at least up to the S&H stage, as the integration time depends on the input capacitance, which has less mismatch than other parameters of the circuit. Also, integration principally acts as a low-pass filter which removes the high frequency components of the noise.

The main disadvantages of this method are the effect of digital noise from the reset and S&H switching transistors, and the inability to change the integration time locally, which means that the dynamic range for a specific integration time is always limited to a certain global value. However, at the expense of area, control circuits can be included at each pixel to vary the integration time [Chen and Ginosar 95].

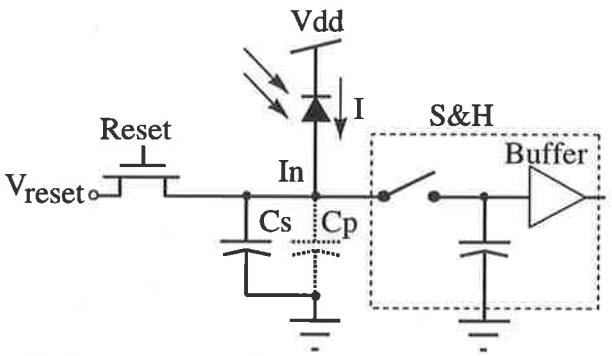


Figure 5.10: Integration based photocircuit.

5.3 Circuits for Spatial Processing

Image information in the spatial domain is an inherent feature of image processing. Circuits for performing spatial image processing are therefore of importance to the design of vision chips. A physical view of spatial image processing is illustrated in Figure 5.11. The processing element at each pixel interacts with its neighbors.

The type of processing, and the number of neighboring cells involved in the processing at each pixel, depend on the specific algorithm being implemented.

In an algorithm which involves all the cells in a neighborhood of N, and assuming that the interconnects only transfer information in one direction, $M_1 = 2((2N+1)^2 - 1)$ interconnections are needed. For reciprocal interconnections this becomes $M_2 = (2N+1)^2 - 1$. Even for $N = 1$ the number of interconnections ($M_1 = 16$) are relatively large.

In some algorithms, only the elements on the main axes are involved, and hence $M_1 = 2{\times}4N$ and $M_2 = 4N$, for uni-directional and bidirectional interconnections, respectively. Even for these types of algorithms the number of wires required is still high ($M_1 = 8$ for $N = 1$).

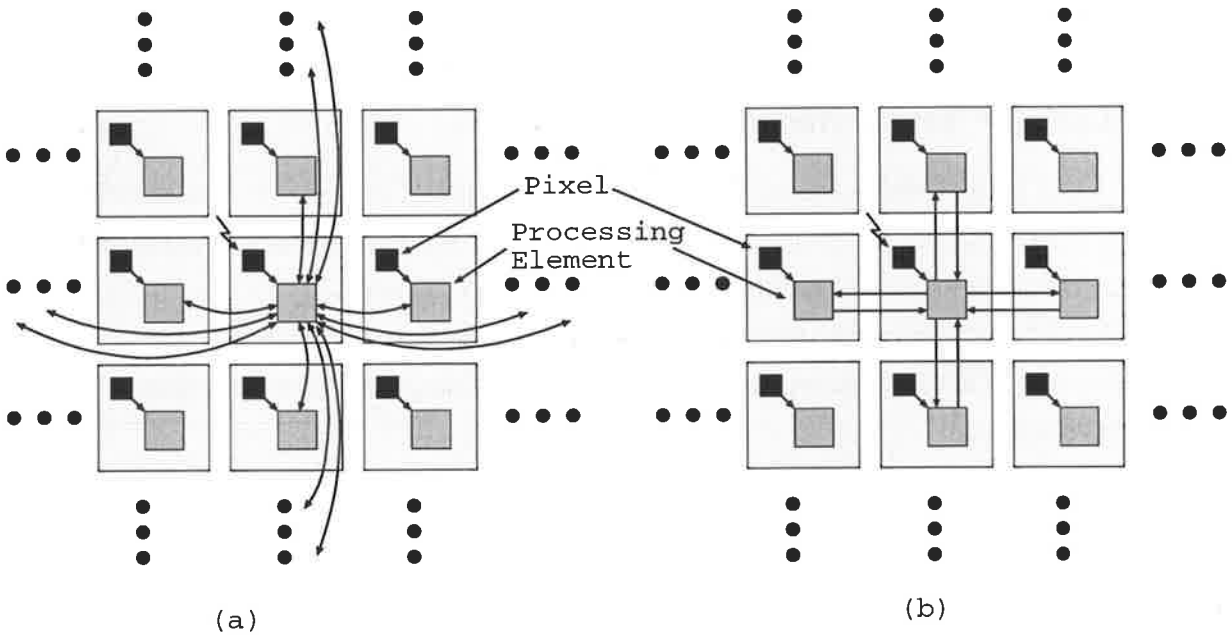


Figure 5.11: A physical view of spatial image processing. a) A general structure with bidirectional interconnects. b) CNN structure with uni-directional and main-axes-only interconnects.

It is therefore obvious that performing any type of spatial image processing, where more than the first nearest neighbors are involved, will be highly interconnect-limited. The cellular neural network (CNN) structure, shown in Figure 5.11-b, in which the only interactions are with the nearest neighbors, presents the most economical architecture for implementing spatial image

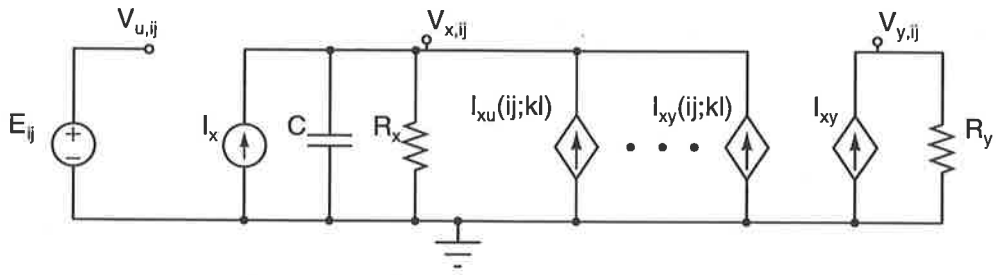


Figure 5.12: A cell circuit in a CNN.

processing chips. In fact CNNs, were originally introduced with an emphasis on being VLSI friendly [Chua and Yang 88a].

CNN and resistive networks are two specially designed networks for performing image processing tasks, and offer a straightforward architecture for VLSI implementation. In the following sections we first review CNN and resistive networks, and then present a *circuit design perspective* of two of the most common image processing operations implemented in VLSI, i.e. spatial smoothing, and retinal-function processing.

5.3.1 Cellular Neural Networks

A cell in the original CNN is based on an analog circuit shown in Figure 5.12 (see [Chua and Yang 88a, Chua and Yang 88b]). E_{ij} is the input voltage, I_x is an independent current source, C and R_x are the capacitance and resistance at the cell, $V_{u,ij}$, $V_{x,ij}$, and $V_{y,ij}$ are the cell input, cell state, and cell output voltages, respectively. $I_{xu}(ij;kl)$ and $I_{xy}(ij;kl)$ are voltage controlled current sources (VCCS) which depend on the cell state and output of the neighboring cells, and can be expressed as:

$$\begin{aligned} I_{xu}(ij;kl) &= B(ij;kl)V_{u,kl} \\ I_{xy}(ij;kl) &= A(ij;kl)V_{y,kl} \end{aligned} \tag{5.5}$$

I_{xy} is a *sigmoid* (bounded monotonically increasing nonlinear) VCCS which is controlled by the cell state voltage $V_{x,ij}$. **A** and **B** are matrices which constitute so-called *templates*. A linear CNN is fully characterized by these two matrices and the constant I_x.

As mathematical treatment of CNNs can be found in the literature [Goles 90], they are considered here only from a VLSI implementation point of view. From the structure of a basic cell, it is clear that for a general CNN with only the first nearest neighbors, 16 interconnections, 19 weight storage elements (to store the A and B matrices), and 19 VCCS elements are required. Each cell should also have a capacitor and resistor for the state node $V_{u,ij}$, and a sigmoid VCCS and a resistor for the output stage. For a fully symmetrical CNN, only 3 weight storage elements are required. Although most of these elements can be realized relatively easily, the required area

for a full implementation has restricted the implementation of large CNN arrays. For example, in a 0.8μm CMOS process, and using traditional circuit design techniques, a cell density of 27 cells/mm^2 is achievable [Dominguez-Castro et al. 97]. About 50% of the area is dedicated to the synapses (weight storage and multiplication).

The programmability of CNNs gives them an advantage. Various image processing operations can be performed using the same hardware by changing the weights in the CNN. This feature would be desirable only when used in an *analogic* fashion (a mixture of analog and logic circuits) [Roska and Chua 93], so that a sequence of image processing tasks are performed using the CNN. In case a single image processing task, such as edge detection, is to be performed using a hardware CNN, then a large proportion of the circuits would be redundant and the hardware would be extremely under-utilized. In this case special purpose vision chips would use the silicon area in a more efficient way. Other factors such as circuit mismatch, cell density, and power dissipation would also favor a special purpose design, rather than a general purpose CNN. For example, a typical VLSI CNN uses more than 100 transistors for each cell [Dominguez-Castro et al. 97]. By contrast, a special purpose design for image smoothing requires only two transistors per cell (see section 5.3.3), and dissipates several orders of magnitude less power.

In addition to representing a specific class of neural networks, the CNN paradigm can also provide a framework for *analyzing* different existing algorithms and networks. For example, the resistive networks described in the following section have been treated using CNN formulations [Shi and Chua 92]. Of course, these algorithms or networks should first be expressed as a CNN, and the associated templates be determined.

5.3.2 Resistive Networks

Resistive networks have long been used for performing spatial image processing [Fukushima et al. 70]. A resistive network using positive-only resistors can perform spatial smoothing on a one or two-dimensional image. In a more general case both negative and positive resistors can be used, and functions other than simple smoothing can be obtained. Resistive networks will be given a more comprehensive treatment, due to their wide usage in a large variety of vision chips designed to date.

A generic second-order uniform resistive network is illustrated in Figure 5.13, in which each node is connected to the first and second nearest neighbors by resistors. The inputs to the network are currents $x(i)$ and the outputs are the node voltages $y(i)$. All resistors are assumed to be linear. The input/output transfer function of the network can be written as

$$x(n) = -g_2 y(n-2) - g_1 y(n-1) + (g_0 + 2g_1 + 2g_2)y(n) - g_1 y(n+1) - g_2 y(n+2) \quad (5.6)$$

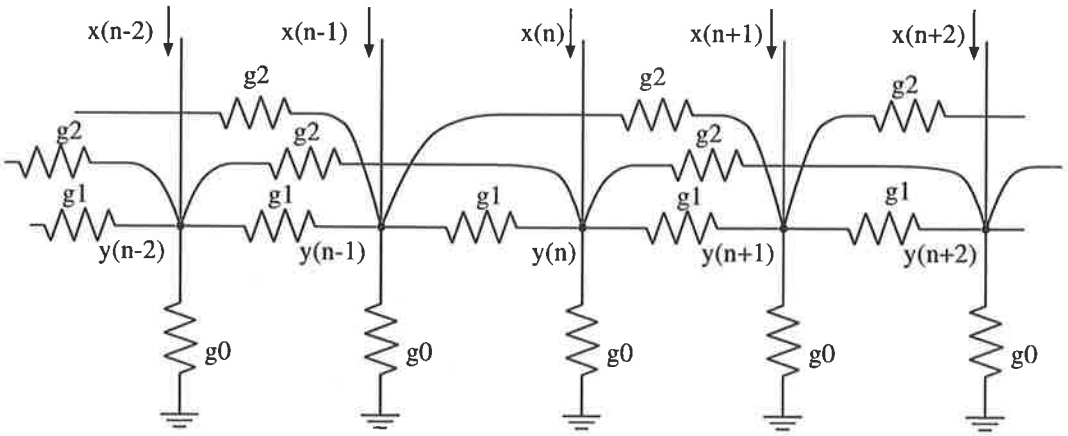


Figure 5.13: Second-order resistive network.

Assuming that the Z-transform of the network exists, it can be expressed as:

$$H(z) = \frac{1}{\sum_{k=-2}^{+2} a_k z^k} \qquad (5.7)$$

$$a_{-2} = a_2 = -g_2 \quad a_{-1} = a_1 = -g_1 \quad a_0 = g_0 + 2g_1 + 2g_2$$

Stability

The conductances g_1 and g_2 can be either positive or negative. However, when negative conductances are used, the network may become unstable (see [Matsumoto et al. 93, Shi and Chua 92, Raffo 96] for various issues pertaining to the stability of resistive networks). For the second-order network, the stability conditions are given by [Matsumoto et al. 93]

$$\begin{cases} -g_0 - 2g_1 + 2|g_1| < 0 & \text{when } g_2 > 0 \text{ or} \quad (a) \\ & g_2 < 0 \text{ and } |g_1/g_2| \geq 4 \quad (b) \\ -g_0 - 2g_1 - 4g_2 - g_1^2/4g_2 < 0 & \text{when } g_2 < 0 \text{ and } |g_1/g_2| \leq 4 \quad (c) \end{cases} \qquad (5.8)$$

In Appendix B, another method based on the analysis of the transfer function of the system is presented. In addition to being straightforward, this method can also lead to the derivation of the characteristics of the kernel function. Figure 5.14 shows the region of stability of the network in the $u = g_1/g_0$, $v = g_2/g_0$ space, assuming that g_0 is positive. The regions 1, 2, 3, and 4 are associated with the first condition in equation 5.8, and region 5 is associated with the second condition.

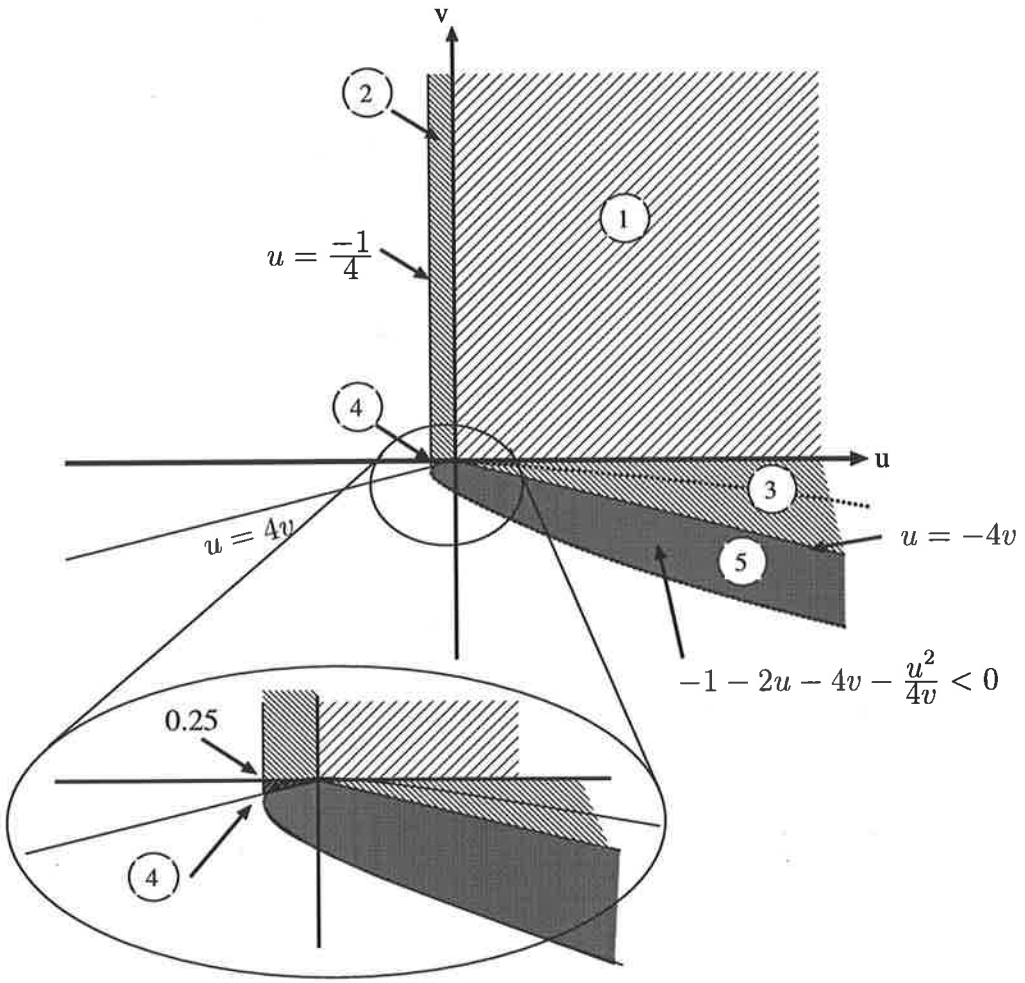


Figure 5.14: Regions of stability for the second-order resistive network in the $u = g_1/g_0$ and $v = g_2/g_0$ space.

Spatial Response

The impulse response of the network, or the convolution kernel realized using the network, can be uniquely determined in the stable regions.

Table 5.2 summarizes the types of kernel functions that are realizable using the second-order network. Refer to Figure B.3 in Appendix B for the impulse response in different regions of the (u, v) space.

Table 5.2: Kernel functions implementable using a second-order resistive network. These are the equations for impulse response for $n > 0$. Note that the function is non-causal, and is symmetric with respect to $n = 0$. Also in the following functions $|\alpha| < 1$ and $|\beta| < 1$.

Function
$h(n) = Ae^{+\alpha n} + Be^{+\beta n}$
$h(n) = Ae^{+\alpha n} - Be^{+\beta n}$
$h(n) = Ae^{+\alpha n} + Be^{-\beta n}$
$h(n)Acos(\omega n)e^{\ \alpha n}$

5.3.3 Circuits and Networks for Spatial Smoothing

Resistive networks are the most intuitive circuits for performing spatial smoothing. In such networks a resistive grid receives the input current, and each node distributes its current among its neighbors. The output can be taken by reading the node voltages. We will see that even the most intricate circuits described in this chapter utilize this simple principle.

A simple spatial smoothing circuit, which uses the principle of current distribution into a resistive network, is illustrated in Figure 5.15. If all the elements in the network have equivalent impedances, as is the case here, the equation relating the input and output currents can be easily derived.

$$\begin{aligned} I_{out}(n) &= \frac{1}{3}[I_{in}(n-1) + I_{in}(n) + I_{in}(n+1)] + \\ &\quad \frac{R_y}{R_x}[I_{out}(n-2) + 2I_{out}(n-1) + 2I_{out}(n+1) + I_{out}(n+2)] \end{aligned} \tag{5.9}$$

This is a linear recursive transfer function. It can be seen that for $R_y \ll R_x$

$$I_{out}(n) = \frac{1}{3}[I_{in}(n-1) + I_{in}(n) + I_{in}(n+1)] \tag{5.10}$$

For $R_y \gg R_x$ all the output nodes are in fact virtually shorted and

$$I_{out}(n) = \frac{1}{N}\sum I_{in} \qquad \forall \quad n \tag{5.11}$$

Implementing such a resistive network would not be economical in standard CMOS processes, because linear passive resistors with large values (to satisfy power consumption constraints) are not readily available, and the smoothing constant of the network cannot be modified.

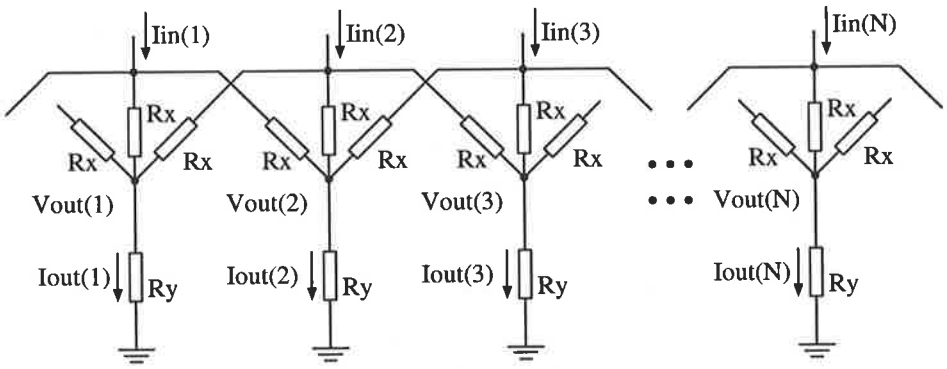


Figure 5.15: Current mode circuit for spatial smoothing.

By replacing the resistive elements with translinear elements (e.g. a junction diode or a MOS diode), a more economical circuit can be realized (see figure 5.16). The expressions describing the function of the circuit can be obtained by applying the translinear principle in the loops indicated by dashed lines, and also the KCL at the input and output nodes of the circuit. It is assumed that all the elements are identical. The derivations can be easily extended for a network with different element values at each branch. Here we only consider this simple case.

$$\begin{aligned} I_{x1} \times I_{out}(1) &= I_{x12} \times I_{out}(2) \\ I_{x2} \times I_{out}(2) &= I_{x21} \times I_{out}(1) \\ I_{x2} \times I_{out}(2) &= I_{x23} \times I_{out}(3) \\ I_{x3} \times I_{out}(3) &= I_{x32} \times I_{out}(2) \end{aligned} \tag{5.12}$$

$$\begin{aligned} I_{in}(2) &= I_{x2} + I_{x21} + I_{x23} \\ I_{out}(2) &= I_{x2} + I_{x12} + I_{x32} \end{aligned} \tag{5.13}$$

A generalized expression can be easily obtained.

$$I_{out}(n) = \frac{I_{in}(n-1)}{\frac{I_{out}(n)}{I_{out}(n-2)} + \frac{I_{out}(n)}{I_{out}(n-1)} + 1} + \frac{I_{in}(n)}{\frac{I_{out}(n)}{I_{out}(n-1)} + 1 + \frac{I_{out}(n)}{I_{out}(n+1)}} + \frac{I_{in}(n+1)}{1 + \frac{I_{out}(n)}{I_{out}(n+1)} + \frac{I_{out}(n)}{I_{out}(n+2)}} \tag{5.14}$$

Although this is a nonlinear recursive equation, the network exhibits a near perfect averaging function similar to a rectangular smoothing window spreading over three neighboring inputs. Figure 5.17 shows the impulse response of this network compared to that of a perfect averaging window.

It can be observed that there is an inhibitory effect demonstrated by the small ringing at the edges of the output. This effect is more dominant for lower contrast inputs. The difference

between the output of this translinear network and the ideal smoothing window is less than 10%.

A drawback of this circuit is the fixed width of the smoothing operation. The network shown in Figure 5.18 achieves a wider smoothing window by using another stage of the current distribution network. Also, the middle branch of each stage can be bypassed by a MOS transistor acting as a switch. The smoothing window of this new network can be adjusted to five, three, or zero (no smoothing operation). For a two-stage smoothing network, the shape of the window is triangular and the input-output relationship can be approximately expressed by

$$I_{out}(n) = I_{in}(n-2) + I_{in}(n-1) + I_{in}(n) + I_{in}(n+1) + I_{in}(n+2) \tag{5.15}$$

This circuit has been used for realizing the multiplicative noise cancellation (MNC) operation in the second motion detection chip, which will be described in chapter 8.

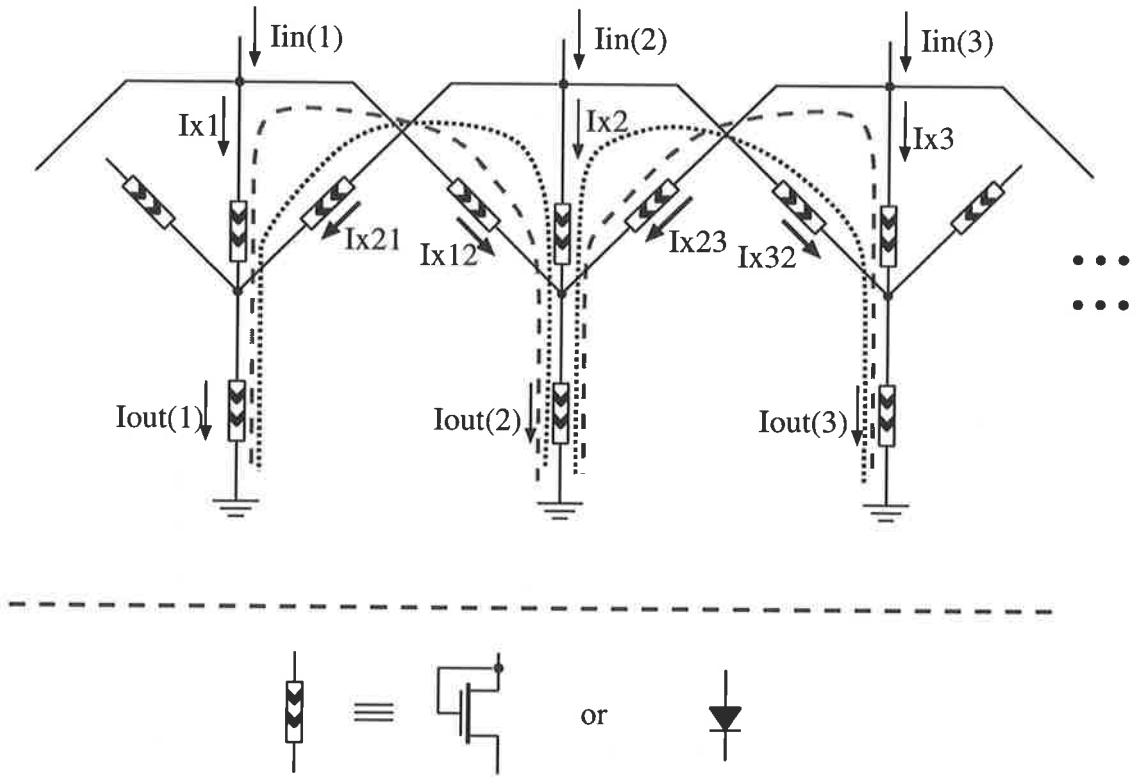
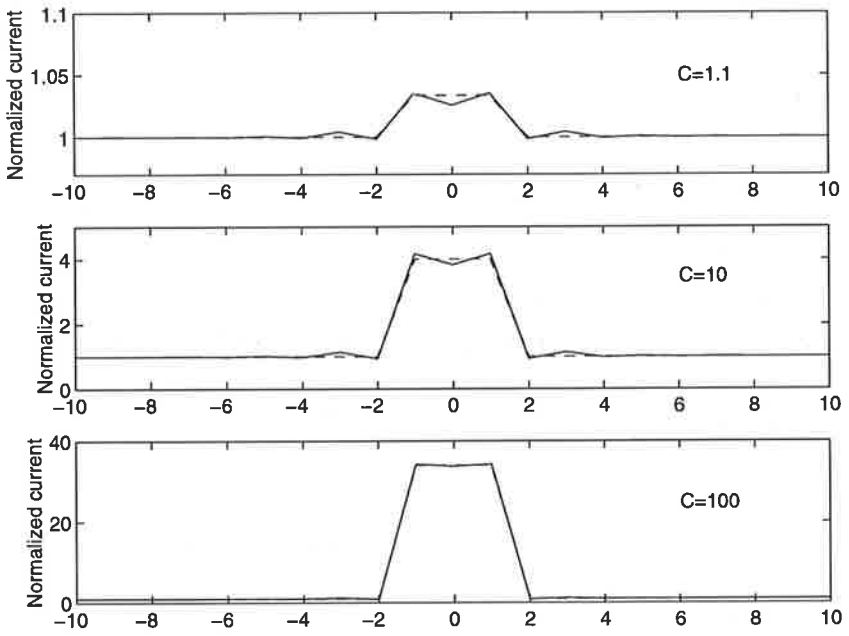


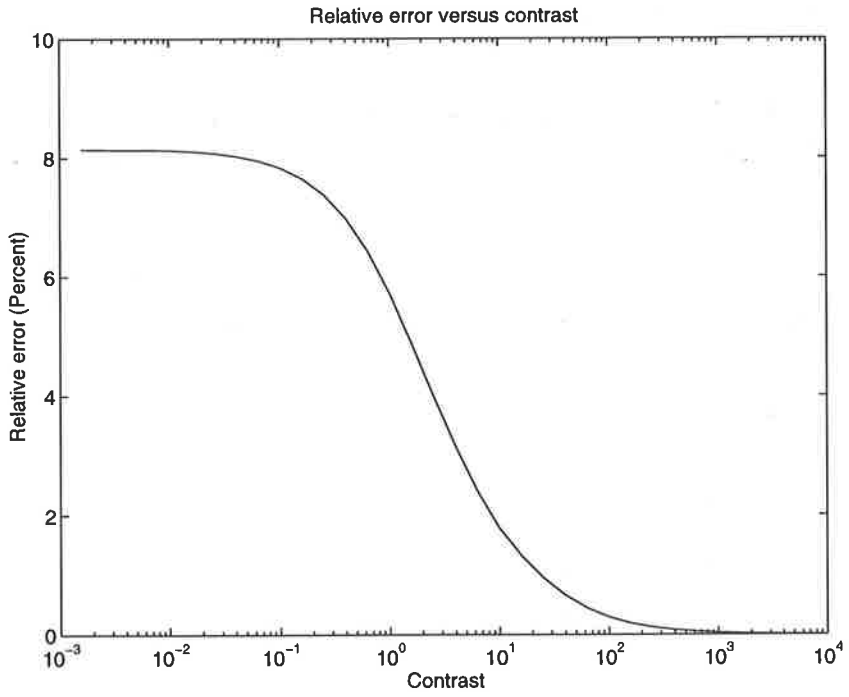
Figure 5.16: A translinear circuit for spatial smoothing.

This circuit can be modified so that the shape of the smoothing window can be adjusted by varying some bias voltages in the circuit, as illustrated in Figure 5.19. Here, the transconductance of the transistors is controlled by the gate voltage. The relationship between the output and input currents can be expressed as

$$I_{out}(n) = \frac{KI_{in}(n-1)}{2K+1} + \frac{I_{in}(n)}{2K+1} + \frac{KI_{in}(n+1)}{2K+1} \tag{5.16}$$



(a)



(b)

Figure 5.17: a) Impulse response of the simple translinear spatial smoothing network. **C** is the height of the input pulse. The rest of the inputs have a value of "1". The dashed lines are from the ideal rectangular averaging window with a width of three. b) Maximum relative error of the impulse response with respect to the ideal averaging window. The x-axis is the contrast of the input at the center and surrounding cells.

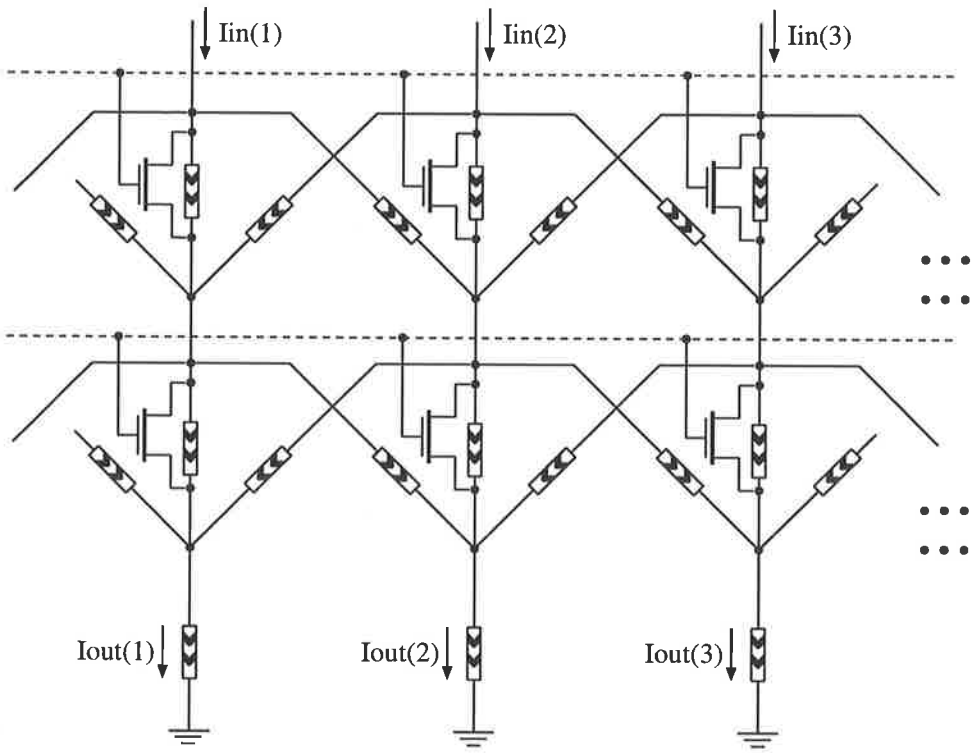


Figure 5.18: Two-stage translinear spatial smoothing circuit.

where $K = e^{\frac{V_r - V_c}{nU_T}}$, n is a process dependent parameter, and $U_T = kT/q$.

Two stages of the smoothing circuit can be used as shown in Figure 5.18, in order to obtain an adjustable-shape smoothing window covering five neighboring cells.

The main drawback of the smoothing circuits described so far is their fixed window size. A slightly modified version of the translinear circuit of Figure 5.19 is shown in Figure 5.20 (see [Andreou et al. 91, Andreou and Boahen 94, Andreou and Boahen 96]). Note that the horizontal transistors operate in the ohmic region. This circuit can be analyzed using the translinear principle by decomposing the horizontal transistors into two back-to-back transistors operating in the saturation region (see Figure 5.21). By writing the translinear equations in the loops marked by dashed lines (note that the loops end at constant voltages V_r and V_c), and the KCL at the circuit nodes we have

$$\begin{aligned} I_{in}(2) &= I_{out}(2) + I_{x32} - I_{x23} + I_{x12} - I_{x21} \\ I_{x32} &= KI_{out}(2) \\ I_{x12} &= KI_{out}(2) \\ I_{x21} &= KI_{out}(1) \\ I_{x23} &= KI_{out}(3) \end{aligned} \tag{5.17}$$

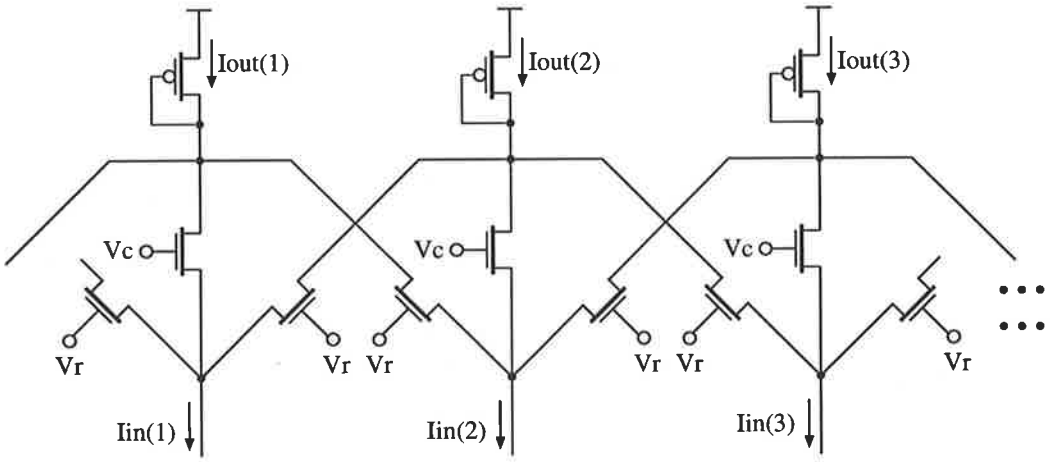


Figure 5.19: Translinear spatial smoothing circuit with adjustable window shape.

where $K = e^{\frac{V_r - V_c}{nU_T}}$. Hence,

$$K\Delta^2 I_{out} + I_{out} = I_{in} \tag{5.18}$$

where Δ^2 is the second spatial-derivative operator. The impulse response of this network is an exponentially decaying function with a decay rate of $1/\sqrt{K}$. Note that this circuit cannot be implemented using bipolar transistors, as the horizontal MOS transistors operate in the ohmic region. All the previous circuits can be implemented using bipolar transistors without any modification to the structure of the networks.

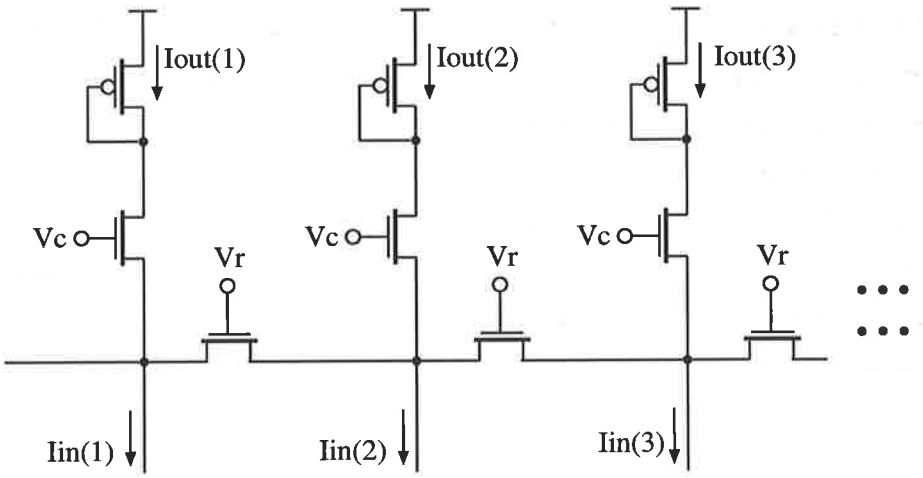


Figure 5.20: Spatial smoothing circuit with adjustable window width.

Figure 5.22 shows another circuit for spatial smoothing, which is an implementation of a first-order resistive network, and was first used for implementing a silicon retina [Mead 89, Mead and Mahowald 88]. The heart of the network is the "horizontal resistor", or HRES, which simulates a floating resistor. The OTA-like circuits are used to bias the two horizontal

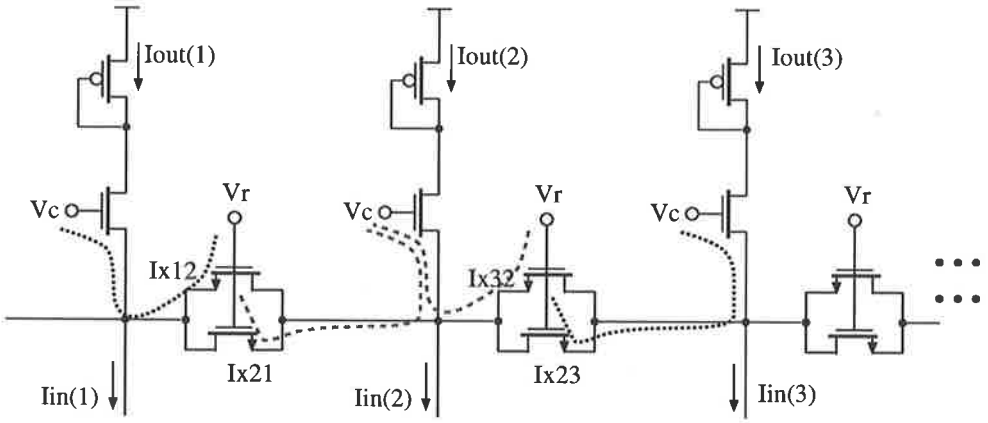


Figure 5.21: Translinear analysis of the smoothing circuit by decomposing the horizontal transistors into two back-to-back transistors.

transistors. Assuming that some of the circuits are shared between neighboring cells, each cell uses 12 transistors. The circuit designed by Andreou and Boahen [Andreou et al. 91, Andreou and Boahen 94] only uses 2 transistors for each cell, a dramatic difference in the number of transistors.

While previous circuits were all designed using continuous current and voltage mode techniques, charge-mode circuits can also be designed for spatial image processing.

In CCD processes, where charge packets convey signal information, charge sharing CCD elements can be used to perform spatial smoothing. In the circuit shown in Figure 5.23, the charge in two separate potential wells are mixed together. Then, by increasing the potential at the middle of the joined well, charge is divided into the wells [Dron 93, Dron McIlrath 96]. This operation can be repeated over several clock cycles, and among neighbors in a large array to spread the smoothing operation over a larger area.

A similar concept can be implemented using switched-capacitor (SC) circuits, although in this case, instead of potential wells, capacitors hold the charge. One advantage of using SC circuits is that they can be implemented in standard CMOS processes, which makes it possible to use various circuits and design techniques available for CMOS.

Considering that a resistive element can be implemented using a capacitor and two switches, as shown in Figure 5.24-a, any resistive network (with positive values) can easily be mapped into a SC network. Figure 5.24-b illustrates a first-order resistive network, where the resistor values can be adjusted by changing the clocking rate of the switches, or by using different capacitor sizes [Umminger and Sodini 92]. In SC circuits, capacitors, switches, and clock signals occupy a significant area. However, as the matching of capacitors is better than that of transistors, higher resolution of at least 7 bits is achievable.

In this section, I presented a series of snapshots of how a very simple and intuitive design can be turned into a very useful, powerful, and yet small circuit. Of course, the evolution of the

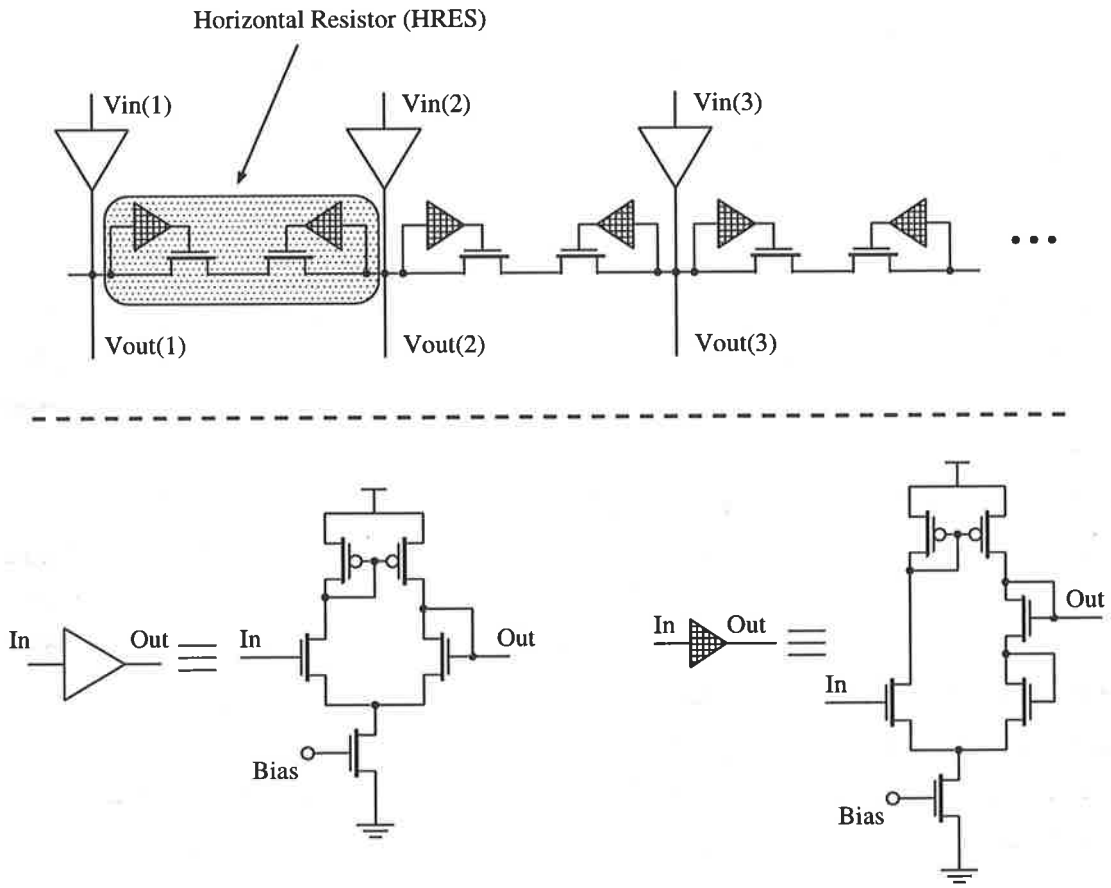


Figure 5.22: Another spatial smoothing circuit, with a large number of transistors. The OTA at each input converts the input voltage to a current and injects it into the network. The other OTA-like circuit (shaded triangle) is used to bias the horizontal transistors.

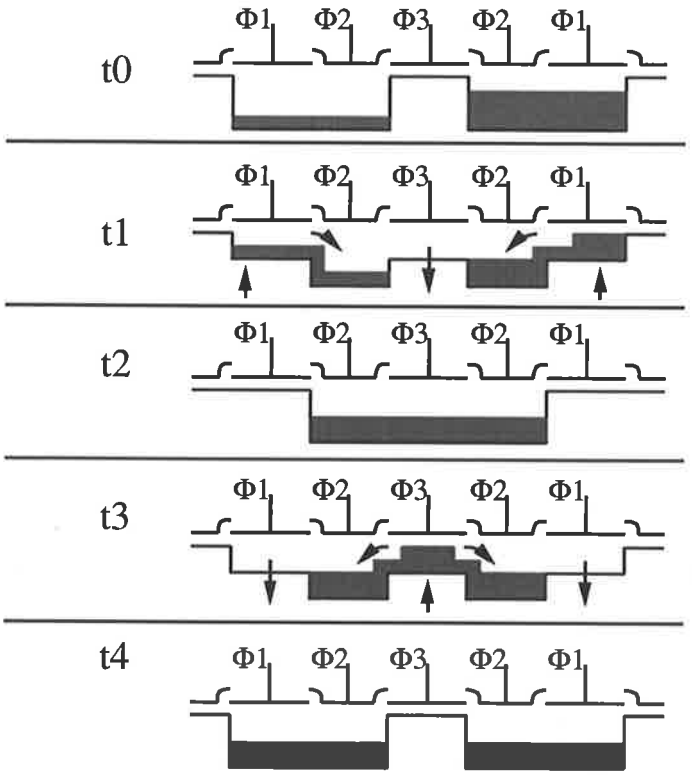


Figure 5.23: The charge redistribution circuit using CCD elements. From top to bottom are the snapshots of the operation.

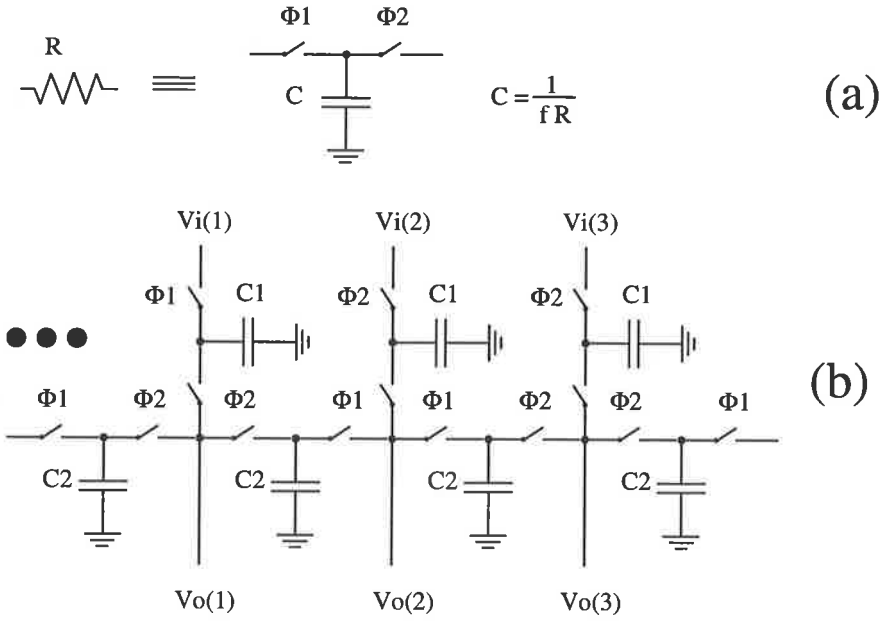


Figure 5.24: a) A resistive element. b) A first-order resistive network using SC circuits.

actual circuits has not followed this straight path, but understanding the principles of the operation of these circuits and the slight variations during this evolution can help in building more complex networks. It has also been tried to apply the translinear principle as much as possible in analyzing the circuits, and there is virtually no mention of complex transistor equations. This also shows how another simple principle can be used as a powerful tool.

5.3.4 Nonlinear Resistive Networks

Nonlinearity provides features which are not readily available in purely linear systems. For example, by adding a simple multiplicative nonlinearity to lateral inhibition models, a new network which is a closer representation of the biological retinal functions can be obtained [Bouzerdoum and Pinter 92]. However, nonlinear systems are more difficult to analyze and synthesize, as illustrated by the difficulty of finding even the DC solution of small circuits comprising nonlinear resistive elements (see for example [Yamamura and Mishina 96,Pastore and Premoli 97]).

Nonlinear resistive networks, as a simple class of general nonlinear periodic networks, can be used in some image processing tasks, for example to avoid extraneous smoothing, to highlight discontinuities, and to discard outliers [Harris et al. 90,Lumsdaine et al. 91, Yu et al. 92,Harris et al. 89,Harris 91]. However, only one type of nonlinear resistive elements, i.e. a resistive fuse, has been used in all these networks. The original drive behind this special nonlinear network has also been intuitive. The characteristics of a resistive fuse network are such that the low contrast parts of the image are smoothed without affecting the high contrast areas, which usually contain more useful information.

Here, only two nonlinear resistive elements will be described: the **saturating** resistive element, and the **resistive fuse**. The general characteristics of these two elements are depicted in Figure 5.25.

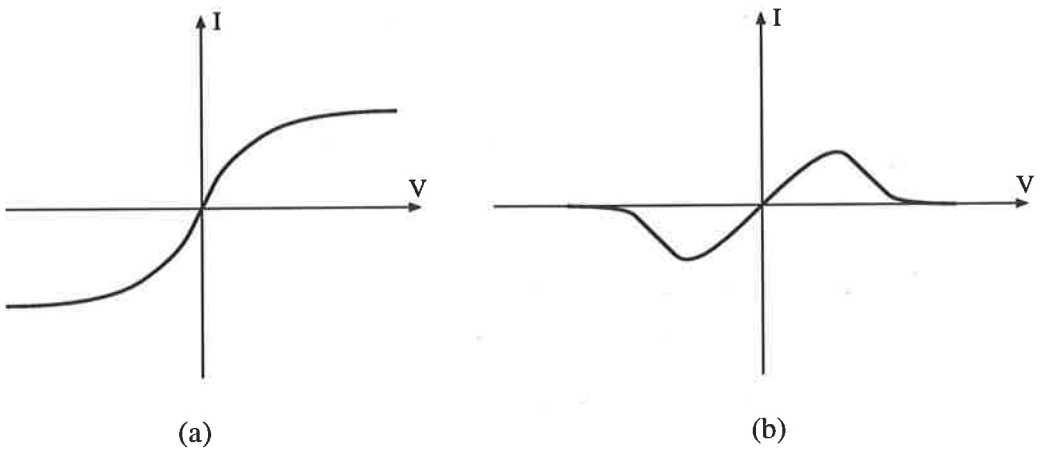


Figure 5.25: I-V characteristics of a) saturating resistive element, b) resistive fuse.

Resistive Networks Using Saturating Resistor

Almost all resistive elements implemented using CMOS transistors exhibit nonlinear I-V characteristics. For example, the horizontal resistor (HRES) shown in Figure 5.22 presents a "hyperbolic tangent" saturating characteristic. For low contrast inputs to the network, the resistive elements behave almost linearly. However, when the voltage across a HRES exceeds the linear range, the current through the resistor saturates and inhibits the spreading of a large contrast. This could be considered as a form of *anti-blooming* effect, whereby very bright spots in the input image are isolated by limiting their effect on their neighbors.

Resistive Networks Using Resistive Fuse

A resistive fuse can be used to accentuate the anti-blooming effect of the saturating resistive element described in the previous section.

In a resistive fuse network, when contrast is very high, the interaction between pixels is virtually absent, and hence sharp contrasts will remain intact, while low contrast parts of the image which have lower signal to noise ratio will be low-pass filtered.

Due to the negative conductance region in the I-V characteristics of this element, however, the network can become unstable, or it may converge to a local minimum state [Harris et al. 90, Harris 91]. A solution to this problem is to bias the resistive elements such that originally they act as linear resistive elements, and gradually change the bias until they behave as resistive fuses. Almost all the resistive fuse circuits proposed are capable of being used in different modes, either as a normal resistive element, or as a resistive fuse [Harris et al. 90, Yu et al. 92].

Several circuits have been proposed for implementing the resistive fuse. The original circuit proposed by J. Harris uses the horizontal resistor (HRES) described in Section 5.3.3. The circuit controls the resistivity by comparing the absolute value of the difference between the voltages at the nodes of the resistive element with a threshold value.

In [Yu et al. 92] another resistive fuse circuit is described. Although this circuit is smaller than Harris', the voltage range over which the circuit is linear is small and cannot be adjusted. However, in Harris' circuit the linear range can be widened by changing the biasing voltages.

5.3.5 Circuits and Networks for Contrast Enhancement

The implementation of some of the contrast enhancement models presented in section 2.3 are discussed here. The models are briefly reintroduced.

1. Subtraction from spatial average

$$I_{out}(n) = I_{in}(n) - \bar{I_{out}}(n) \tag{5.19}$$

2. Division by spatial average

$$I_{out}(n) = \frac{I_{in}(n)}{\bar{I_{out}}(n)} \tag{5.20}$$

3. Laplacian of Gaussian, Difference of Gaussian (DoG), and Difference of Exponentials (DoE)

$$\begin{array}{lll} I_{out} = \nabla^2 G(I_{in}) & & \text{LoG} \\ I_{out} = G_1(I_{in}) - G_2(I_{in}) & & \text{DoG} \\ I_{out} = E_1(I_{in}) - E_2(I_{in}) & & \text{DoE} \end{array} \tag{5.21}$$

4. Even Gabor function

$$I_{out} = I_{in} * (Acos(\alpha x)e^{-\frac{x^2}{\sigma^2}}) \tag{5.22}$$

5. Regularization theory based (solution of the biharmonic equation)

$$\lambda \nabla^2 \nabla^2 I_{out}(x,y) + I_{out}(x,y) = I_{in}(x,y) \tag{5.23}$$

6. Linear and shunting lateral inhibition

$$\begin{array}{lll} \frac{dI_{in}(n)}{dt} = I_{in}(n) - aI_{out}(n) - b(I_{out}(n-1+I_{out}(n+1)) & & \text{LLI} \\ \frac{dI_{in}(n)}{dt} = I_{in}(n) - aI_{out}(n) - bI_{out}(n)(I_{out}(n-1) + I_{out}(n+1)) & & \text{SI} \end{array} \tag{5.24}$$

The first and second models can be implemented easily by using one layer of spatial smoothing followed by a subtraction or division circuit, as shown in Figure 5.26-a.

The LoG or DoG operators require Gaussian smoothing functions, whose implementation require a negative resistive element [Kobayashi et al. 91]. By contrast, the DoE function can be implemented either with two smoothing layers, using a first order resistive network, or with the diffusive circuit in Figure 5.20. The subtraction can be readily performed in the current or voltage domain.

There does not exist a hardware implementation of the even Gabor function. However, an approximation of this function in the form of a cosine modulated exponential can be implemented using a second order resistive network, which uses negative resistors. The range of resistor values that result in this function can be found in Appendix B .

An implementation that approximates the solution to the biharmonic equation has been described by Andreou and Boahen [Boahen and Andreou 92,Andreou and Boahen 94]. In essence, two coupled diffusive networks are used, which produce two coupled difference equations whose input-output relationship is described by the biharmonic equation.

The implementation of linear and shunting lateral inhibition is relatively straightforward, as there is no need to perform a spatial smoothing function, and only subtraction of the output from a weighted sum of the neighbors is required. The "weighted sum" function can be implemented using either a Gilbert multiplier, or a translinear divider/multiplier [Andreou et al. 91]. In shunting inhibition, the output of the neighbors should also be involved in the "weighted sum" product. In both cases, current mode implementations are preferred, due to the simplicity of the current mode addition and subtraction operations. Chapter 10 describes a current mode implementation of shunting inhibition.

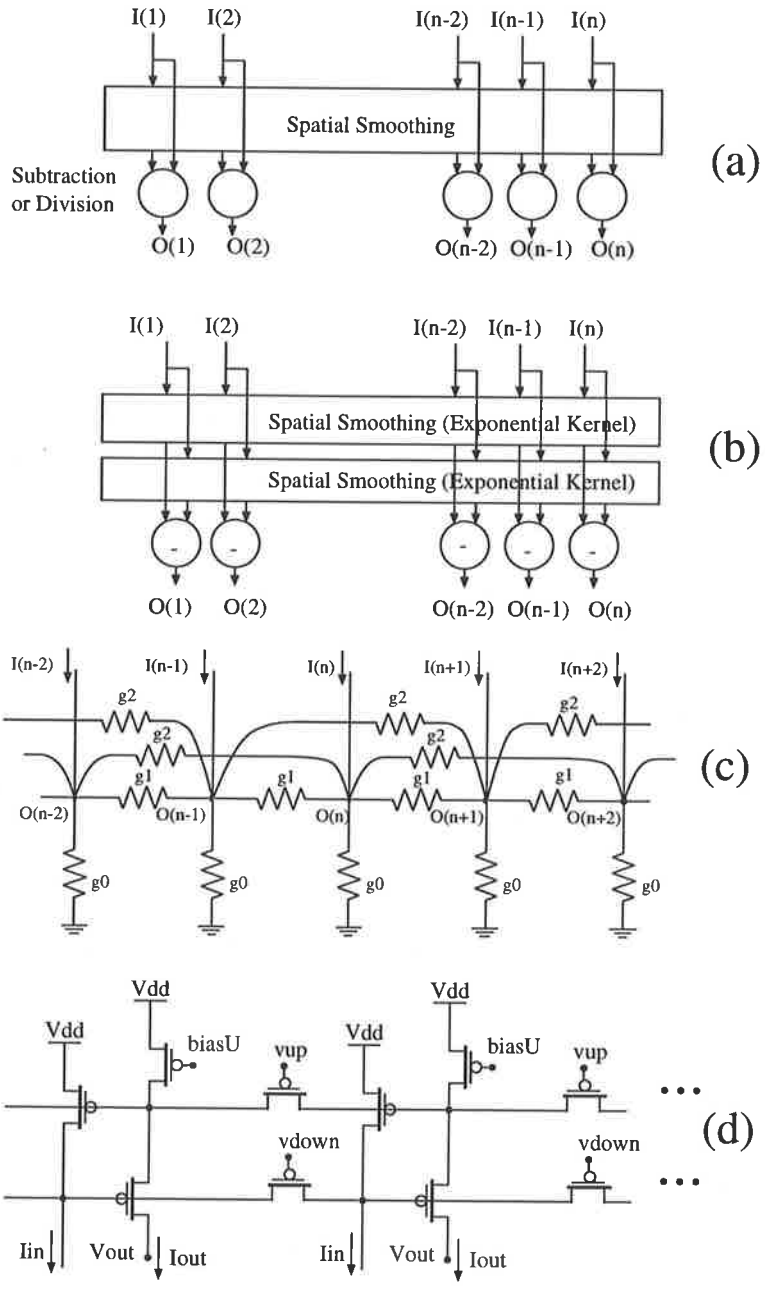


Figure 5.26: Implementations of contrast enhancement.

5.4 Spatio-Temporal Processing

In designing circuits for spatial and temporal image processing, memory elements are required to store or delay image information from the past frames or time sequences. The only practical storage elements in current VLSI technologies are the capacitive storage and delay elements. In a storage element, or analog memory, charge is sampled and stored on a capacitance. By contrast, the signal in a delay element is continuously delayed. The main challenge in designing such elements is to achieve relatively long storage or delay times. In analog memories the storage time is limited by the leakage through unavoidable junctions that are connected to the capacitance, while in delay elements (for example in simple $R-C$ or G_m-C elements) the delay time is limited by the achievable maximum resistance or minimum transconductance.

For vision chip applications, the amount of delay or storage time directly depends on the extent to which past image information is involved in the processing. For example, for a frame differencing operation, where the image is simply subtracted from the previous frame, only one storage element per pixel is required. However, in many motion detection algorithms, several frames may be involved in the processing. Therefore, some of the storage elements require more retention time than the minimum (the time between two frames).

From a circuit design point of view, the only difference between implementing a spatial and spatio-temporal design is the introduction of the delay or storage elements. Therefore, it is necessary to review analog memory elements.

5.4.1 Analog Memory Elements

The simplest structure for charge storage is the sample & hold cell shown in Figure 5.27. The leakage current of the source of the switch transistor limits the charge retention time, which may be up to a few seconds for digital signal storage. The acceptable retention time for an analog application obviously depends on the resolution required, but drops from that for DRAM cells by two orders of magnitude (for 8 bits of resolution by 1/256). This circuit is useful only for very short term storage, such as in small imagers with fast frame rates.

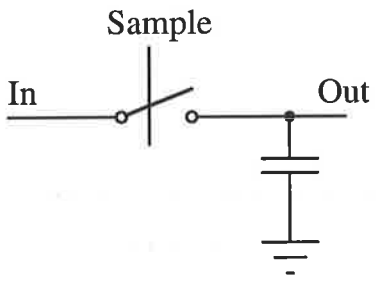


Figure 5.27: Sample & hold analog memory

The storage time of the cell can be increased by using several techniques, such as differential storage, and leakage reduction. In the differential storage technique, the original signal is translated into a differential signal and stored on two similar storage devices. As the leakage reduces the charge at both nodes at almost the same rate, the difference remains the same. A drawback of this technique is the additional area consumed by single ended-to-differential translation and the extra capacitance.

In the leakage reduction technique, the leakage of the source/drain diffusion of the switching transistor at the storage node is reduced by setting the voltage across the two nodes of the source diffusion-well diode to zero [Vittoz et al. 91]. Using the circuit shown in Figure 5.28, storage times of up to several seconds can be achieved under normal conditions.

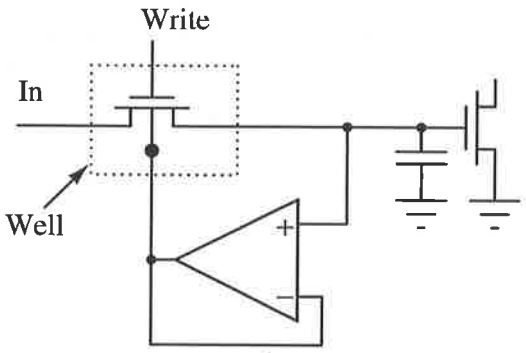


Figure 5.28: Circuit for reducing the leakage.

Floating gate structures have long been used in digital EPROM devices, as well as in many implementations of analog circuits and systems. Despite the long term storage achieved with floating gate structures (in the order of several years), the accuracy, programming, and reprogramming issues of these devices remain challenging. Floating gate devices can be found either in special processes, where thin-gate devices are available for low voltage programming, or in standard processes, where the gate of a normal transistor is left floating. The floating gate devices in standard processes require either high programming voltages, which might exceed the breakdown voltage of different junctions in the process, or accelerated mechanisms such as exposure to UV light.

5.4.2 Continuous Delay Elements

A continuous delay element is often approximated by circuits, such as integrators. Figures 5.29-a and 5.29-b show two basic voltage mode circuits for delaying signals. Both circuits add distortion in phase and amplitude to the input signal. However, in many motion detection applications these sources of distortion can be tolerated, and more accurate delay elements are unnecessary. The current mode delay element in Figure 5.29-c has been used as an essential

building block in current mode circuits[1]. Alternatively, a simple log-domain integrator, shown in Figure 5.29-d, can also be used as a delay element [Seevinck 90].

The amount of delay in the RC network depends on the resistor value, in the OTA-C circuit it depends on the bias current, and in the current-mode delay element it depends on the input current level.

In order to achieve large delay times using a conventional OTA-C circuit, very small biasing currents are required. This causes several problems, including increased mismatch at low current levels, and sensitivity to different noise sources. This requirement can be reduced by using linearization techniques applied to the OTA [Furth and Andreou 95, Moini et al. 97b].

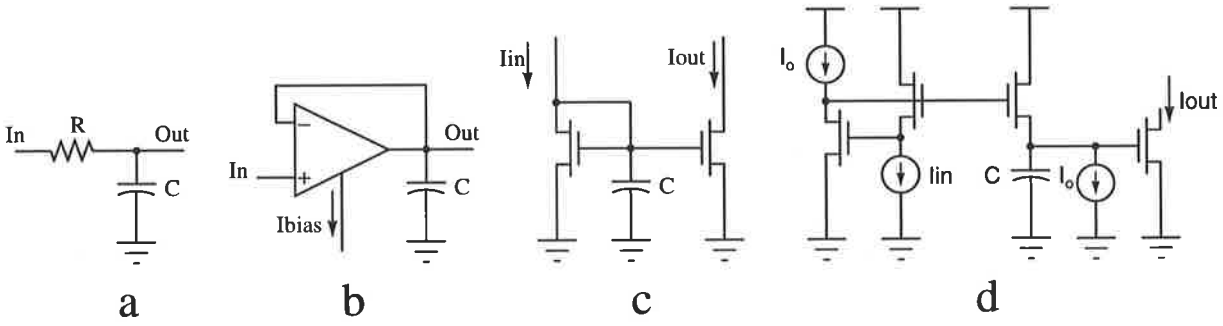


Figure 5.29: a) a RC circuit used as a delay element, b) an OTA-C circuit as a delay element, c) a current mode delay element. d) log-domain integrator.

5.5 Summary

This chapter presented an overview on the devices, circuits, and networks required in vision chips. Photodetector elements and their spectral response were discussed in Section 5.1. Various types of photocircuits, which transduce the photocurrent into a voltage were described in Section 5.2. Circuits and networks for spatial and spatio-temporal image processing were presented in Sections 5.3 and 5.4.

[1]Dynamic current mirrors are an example of sampled data storage element. However, the principal function of these circuits is to store the gate voltage. Therefore, the storage is essentially working in voltage mode.

Chapter 6

Issues

Although the field of analog VLSI has led to some understanding of the function of biological systems, their use and progress has been largely hampered by problems that will be discussed in the following sections.

6.1 Mismatch

Mismatch in transistors has long been recognized as one of the barriers to achieving high precision in analog circuits. Current research on CMOS imagers has contributed to the study of mismatch, which is known as fixed pattern noise (FPN) in the imager terminology. However, most of these studies have been conducted on the mismatch in a single transistor [Shyu et al. 84, Lakshmikumar et al. 86, Pelgrom et al. 89, Forti and Wright 94, Pavasovic et al. 94, Steyaert et al. 94, Bastos et al. 95, Bastos et al. 97]. The effect of mismatch in interconnected networks of interest to us, such as silicon retinas, has been largely ignored, or studied on very simple or special cases [Shi and Chua 92, Kinget and Steyaert 96].

The performance of a circuit in the presence of mismatch cannot easily be derived even for very simple operations using ideal linear networks, as it not only depends on the general function of the network, but also on the detailed implementation of the circuits. For example, even for a first-order resistive network, the algebra required to characterize the effect of mismatch is rather involved [Shi and Chua 92]. The study by Shi and Chua illustrates that the mismatch at the output of the network also depends on the spatial contents of the input. This is a very fundamental difference between interconnected networks and isolated circuits. As we will see, mismatch in the output of contrast enhancement circuits is highest when the spatial content of the input image is high.

In this section a brief background on the mismatch problem, its sources, and modeling, is presented. Then simulation and test results on the effect of mismatch on analog circuits and networks are presented. Current mirrors are used as an example in analyzing the mismatch

properties of circuits.

6.1.1 Sources and Modeling of Mismatch

Random mismatch is caused by variations in geometrical or electrical parameters of the device. Variations in the width and length of a MOS transistor directly translate into variation of the transistor conductance. These geometrical variations, however, have less effect than electrical or physical parameters, as they can be more finely controlled, and the use of larger devices reduces the relative variations.

Electrical mismatch is mainly due to two parameters, the threshold voltage V_T and the conductance β. The threshold voltage mismatch depends on the charge distribution in the channel and oxide, and can be expressed as:

$$V_T = \phi_{MS} + 2\phi_B + \frac{Q_B}{C} - \frac{Q_F}{C} + \frac{qD_I}{C} \tag{6.1}$$

where ϕ_{MS} is the difference between the work function of the gate material and the semiconductor in the channel, ϕ_B is the Fermi potential of the semiconductor, Q_F is the fixed charge density in the oxide, Q_B is the charge density in the depletion region, D_I is the doping density for adjusting the threshold voltage, and C is the unit-area capacitance of the gate oxide.

The variance of threshold voltage, σ_{VT}, can be written as:

$$\sigma_{VT}^2 = \frac{1}{C^2}(\sigma_{QB}^2 + \sigma_{QF}^2 + q^2\sigma_{DI}^2) + \frac{\sigma_C^2}{C^2}\left(\frac{Q_B^2}{C^2} + \frac{Q_F^2}{C^2} + \frac{q^2D_I^2}{C^2}\right) \tag{6.2}$$

Assuming that the probability distribution function of all the charges is a Poisson distribution [Lakshmikumar et al. 86], this can be written as

$$\sigma_{VT}^2 = \frac{q}{LWC^2}(Q_B^2 + Q_F^2 + qD_I) + \frac{\sigma_C^2}{C^2}\left(\frac{Q_B^2}{C^2} + \frac{Q_F^2}{C^2} + \frac{q^2D_I^2}{C^2}\right) \tag{6.3}$$

The second set of terms in the right hand side indicate the capacitance variations, which are due to geometrical variations and are often much less than the variations due to the first set of terms. From the first set of terms the Q_B and qD_I components have been found to be dominant.

The variance of the conductance of the transistor in the strong inversion region can be expressed as:

$$\begin{aligned} \beta &= \mu C_{ox}\frac{W}{L} \\ \frac{\Delta\beta}{\beta} &= \frac{\Delta\mu}{\mu} + \frac{\Delta C_{ox}}{C_{ox}} + \frac{\Delta W}{W} - \frac{\Delta L}{L} \\ \frac{\sigma_{\Delta\beta}^2}{\beta^2} &= \frac{\sigma_{\Delta\mu}^2}{\mu^2} + \frac{\sigma_{\Delta C_{ox}}^2}{C_{ox}^2} + \frac{\sigma_{\Delta W}^2}{W^2} + \frac{\sigma_{\Delta L}^2}{L^2} \end{aligned} \tag{6.4}$$

The last three terms all relate to the geometrical variations. In N-channel MOSFETs, the variation of L and W are the main source of mismatch of the transistor conductance. For P-channel transistors, it is believed that the poorer matching of the gate oxide capacitance C_{ox} contributes to the majority of the observed mismatch.

Notice that mismatch in the threshold voltage and conductance both depend on the variations of W, L, and C_{ox}. Therefore, the variances of these two parameters are not statistically independent. Principle component analysis (PCA) can be used to find the main variational components. However, often it is preferred to relate the variations to more physically meaningful parameters, such as threshold voltage and conductance.

In the weak inversion or subthreshold region, the conductance and its variation are given by

$$\begin{aligned} I_{D0} &= K\mu\frac{W}{L}e^{-\frac{V_{T0}}{nU_T}} \\ \frac{\Delta I_{D0}}{I_{D0}} &= \frac{\Delta\mu}{\mu} + \frac{\Delta W}{W} + \frac{\Delta L}{L} + \frac{\Delta V_{T0}}{V_{T0}} \\ \frac{\sigma^2_{\Delta ID0}}{I^2_{D0}} &= \frac{\sigma^2_{\Delta\mu}}{\mu^2} + \frac{\sigma^2_{\Delta W}}{W^2} + \frac{\sigma^2_{\Delta L}}{L^2} + \frac{\sigma^2_{\Delta VT0}}{n^2U_T^2} \end{aligned} \tag{6.5}$$

where K and n are process dependent parameters; n ranges from 1 to 1.3 .

From the above equations, the effect of the width and length variations on mismatch is clear. By using larger devices, both the geometrical and electrical (charge distribution and mobility) parameters will be better matched. A first order model relates mismatch to the device area LW as follows.

$$\begin{aligned} \frac{\sigma_\beta}{\beta} &\propto \sqrt{\frac{1}{LW}} \\ \frac{\sigma_{V_T}}{V_T} &\propto \sqrt{\frac{1}{LW}} \end{aligned} \tag{6.6}$$

For moderate size transistors, this gives a very good indication about the relative amount of mismatch. For short-channel transistors, the linear relationship between mismatch and area does not hold and alternative models such as that proposed by [Bastos et al. 97] should be used. Although a dependence between mismatch and the distance of devices from each other has been reported, it is observed only in very large transistors. In moderate and small size transistors, mismatch due to area dominates the distance factor [Bastos et al. 97].

6.1.2 Mismatch in Circuits

Characterizing mismatch in circuits by trying to derive closed form equations is cumbersome and in many cases impractical, except for special class of circuits.

Many of the circuits designed in vision chips, are based on the translinear principle. In a simple translinear circuit implemented using MOS transistors operating in the subthreshold region, the output current is

$$I_{out} = \frac{\prod_n \frac{I_n}{I_{Dn}}}{\prod_m \frac{I_m}{I_{Dm}}} \tag{6.7}$$

where I_{Dn} and I_{Dm} are the conductances of the transistors. The variation of the output current can easily be obtained.

$$\frac{\sigma^2_{Iout}}{I^2_{out}} = \sum_{i=1}^{n+m} \frac{\sigma^2_{ID(i)}}{I^2_{D(i)}} \tag{6.8}$$

if we assume that the relative variation of the conductance of all transistors are the same, then

$$\frac{\sigma_{Iout}}{I_{out}} = \sqrt{n+m}\left(\frac{\sigma^2_{ID}}{I^2_D}\right) \tag{6.9}$$

This means that the total relative variance of the output current is proportional to the number of transistors in the translinear loop. In the subthreshold region, the $\sigma I/I$ term is almost independent of the current level [Forti and Wright 94].

Current Mirrors

Current mirrors are the most widely used building block for many types of analog circuits. While mismatch can be reduced by increasing the size of the transistors, this would be impractical for vision chips, where a large number of circuits are to be integrated into a chip, and hence the size of the transistors becomes important.

In order to study the effect of mismatch on current mirrors, four structures have been chosen (Figure 6.1). All of these structures use four transistors, two at the input and two at the output branch. This choice has been made to provide a fair comparison among the structures. The sizes of all transistors are the same, and the substrates are all connected to the ground node. *cm1* is actually a simple current mirror, where the two transistors in each side have been connected in series and can be regarded as a long transistor. The *cm2* current mirror is the cascade mirror, *cm3* is the Wilson current mirror, and *cm4* is a source degenerated current mirror. Table 6.1 shows the relative variance of the output current for each structure. Clearly, *cm1* and *cm4* have about one quarter as much mismatch as the other two structures.

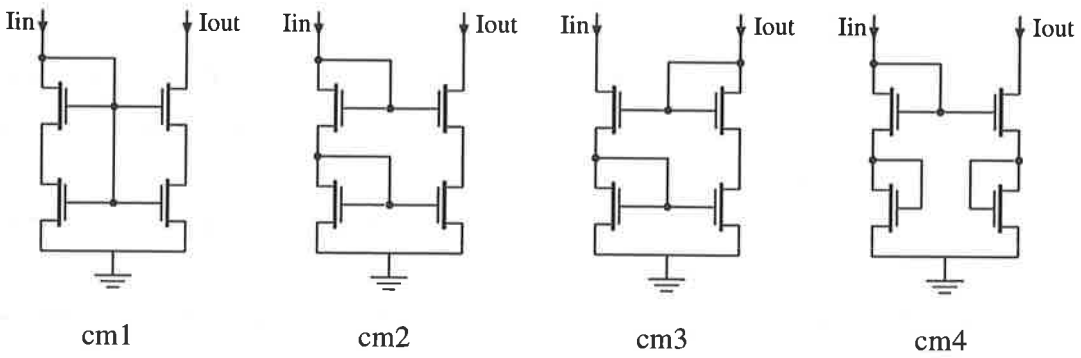


Figure 6.1: Current mirror structures.

Table 6.1: Relative variance of the current mirror structures.

	$\left(\frac{\sigma_{I_{out}}}{I_{out}}\right)^2$
cm1	$\left(\frac{\sigma_{I_{D0}}}{2I_{D0}}\right)^2$
cm2	$\left(\frac{\sigma_{I_{D0}}}{I_{D0}}\right)^2$
cm3	$\left(\frac{\sigma_{I_{D0}}}{I_{D0}}\right)^2$
cm4	$\left[\left(\frac{n}{n+1}\right)^2 + \frac{1}{(n+1)^2}\right]\left(\frac{1}{I_{D0}^2} + \frac{\frac{1}{4n}}{I_{D0}^{\frac{n}{n+1}}}\right)\sigma_{I_{D0}}^2 \approx \left(\frac{\sigma_{I_{D0}}}{2I_{D0}}\right)^2$

We have used Monte Carlo simulations to compare the performance of the current mirrors. Figure 6.2 shows the mean and standard deviation of the output current when the width of the transistors has been varied with a 1% Gaussian distribution. Figure 6.3 shows the same data when the conductance I_{D0} of the transistors is varied. The simulations clearly confirm the analytical expressions. The normalized mean value represents the amount of offset observed in each structure. In the subthreshold region, the offset of *cm1* and *cm4* is less than that of the other two structures, but this is reversed in the strong inversion region.

This simple analysis and simulation demonstrates that, even though the more common current mirrors, i.e. cascade and Wilson, may be superior in terms of measures such as output impedance, other circuits provide better matching.

6.1.3 Mismatch in Interconnected Networks

In this section the effect of mismatch in spatial processing networks is studied. Two types of spatial operation that are common in vision chips, i.e. spatial smoothing and contrast enhancement, are discussed. Deriving analytical expressions is impractical even for the simplest network, which is a spatial smoothing circuit shown in Figure 5.20. When several parameters of the circuit are varied at the same time, the Monte Carlo analysis offers the simplest way in which to characterize the mismatch performance of the network.

The purpose of this section is to demonstrate that the general perception about the mismatch reduction characteristic of silicon retinas is not correct in general, and to show that the effect of mismatch is almost entirely dependent on the specific implementation. Furthermore, the amount of mismatch is shown to be dependent on the spatial content of the image.

In all the simulations, a one-dimensional network with 64 elements is assumed. The bound-

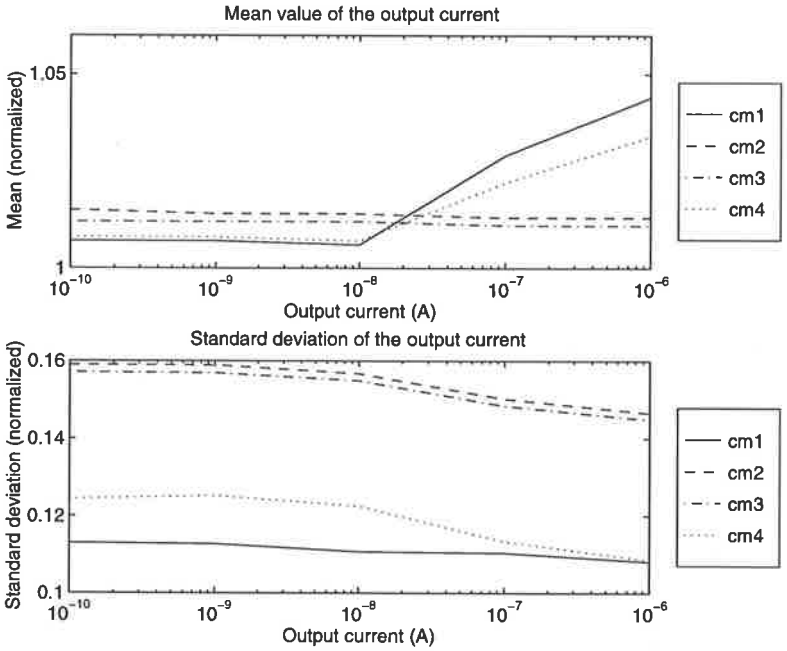


Figure 6.2: Variance and mean of the output distribution with respect to input current, for a 1% Gaussian distribution of the width of all transistors.

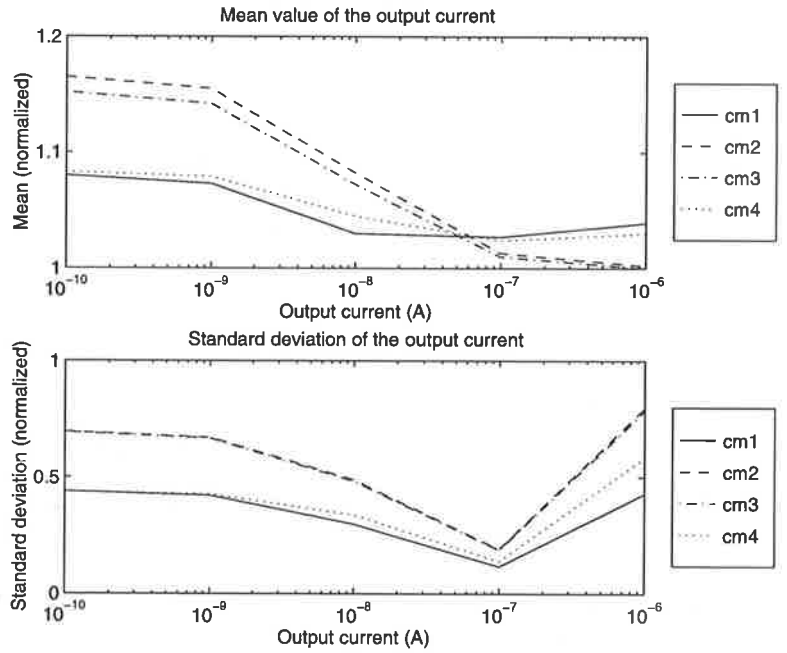


Figure 6.3: Variance and mean of the output distribution with respect to input current, for a 1% Gaussian distribution of the zero-bias threshold voltage of all the transistors.

ary conditions have been chosen to be circular, i.e. the last node is connected back to the first node. The input also has a form of

$$I_{in}(n) = I_0 + I_A sin(\frac{2\pi F}{64} n) \qquad \text{for } F \in [0 \ldots 31] \tag{6.10}$$

This choice of the spatial input and the circular boundary produces a virtual infinite network with a sinusoidal input.

Spatial Smoothing

The first circuit which we analyze is the spatial smoothing circuit shown in Figure 6.4.

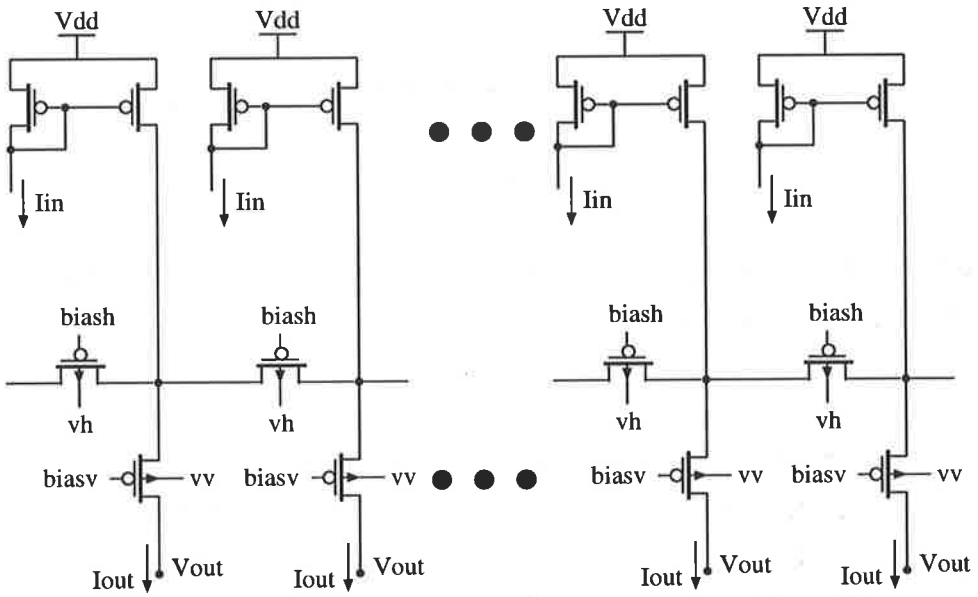


Figure 6.4: Spatial smoothing circuit.

In the Monte Carlo analysis, the threshold voltage of each transistor has been independently varied. The distribution of the threshold voltage is assumed to be Gaussian with a 3σ value of 10 mV, which has been chosen from experimental results and according to the size of the transistors [Forti and Wright 94, Bastos et al. 97]. This amount of mismatch in a current mirror results in a 3σ variance of about 30% of the nominal value of the current.

Figure 6.5 shows the variance of the output current for different spatial frequencies and for different biasing voltages applied to $biasv$ in Figure 6.4, while $biash$ is fixed at 3 volts. When $biasv$ is much smaller than $biash$ ($biasv - biash \ll U_T$) the circuit does not perform any spatial smoothing, and when $biash$ is much smaller than $biasv$, the circuit virtually shorts all the input nodes.

In Figure 6.5-a only the transistors involved in the smoothing are assumed to have mismatch, while in Figure 6.5-b all transistors including the current mirror have mismatch.

Important points from these two results are discussed here.

- When looking along the axis where $biasv$ is varied, the output has the highest variance when it performs a moderate level of spatial smoothing. *This is where it is needed most.* When $biasv$ is very small or very large, the circuit does not perform any *useful* operations. At these extremes the circuit either does not perform any smoothing, or completely smoothes the image (all the horizontal transistors are shorted together).

- Along the spatial frequency axis, output variance is lowest when the input is relatively smooth, and increases when the spatial frequency increases. Again, this means that whenever the input has useful information, i.e. high spatial frequency content, the amount of output variance increases.

- In Figure 6.5-b, where it is assumed that the current mirrors also contribute to the mismatch, the variance is almost completely dominated by the mismatch from the current mirrors. Mismatch in the current mirrors produces *spatial noise*, which is reduced by the spatial smoothing operation. At high values of $biasv$ (shorting all the inputs) the amount of output current variance is about half of that when $biasv$ is small (no smoothing).

Contrast Enhancement

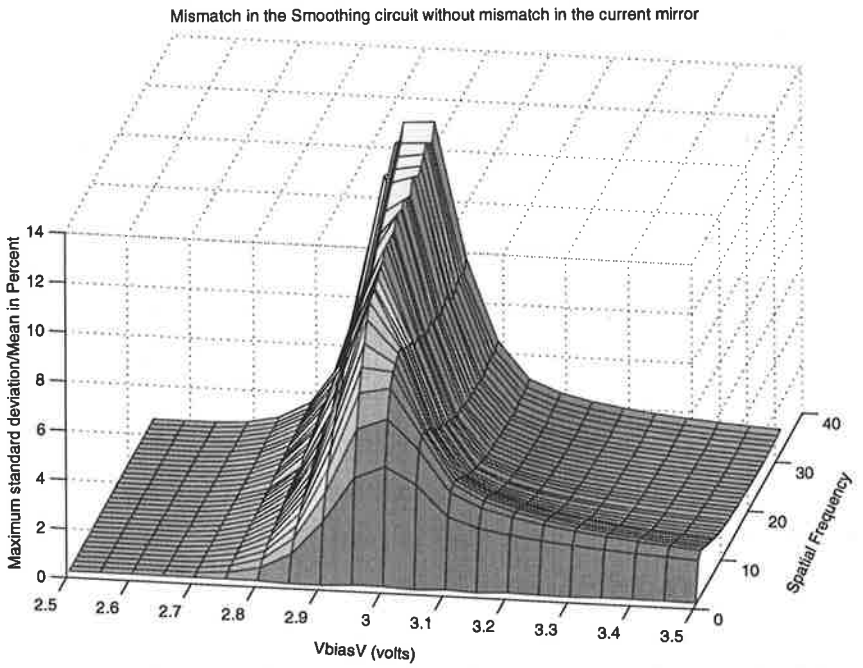
I only present investigations of two implementations of contrast enhancement. The first circuit, which has been extensively used in many vision chips, has been proposed by Andreou and Boahen [Andreou et al. 91, Andreou and Boahen 94, Andreou and Boahen 96], and is shown in Figure 6.6-a. The second circuit is the MNC (multiplicative noise cancellation) circuit (see section 2.3 for more details), which was originally used for reducing the effect of the AC noise from light sources, but also exhibits contrast enhancement properties [Moini et al. 95a, Moini et al. 97b]. The circuit diagram is shown in Figure 6.6-b. In the following discussions the circuits are referred to as AB and MNC.

Figure 6.7-a shows the simulation result for the AB circuit, and Figures 6.7-b,c,d show the results for the MNC circuit. In Figure 6.7-b, all the transistors in the MNC contribute to the mismatch. In Figure 6.7-c the divider is assumed to be free of mismatch, and in Figure 6.7-d only the divider is assumed to have mismatch.

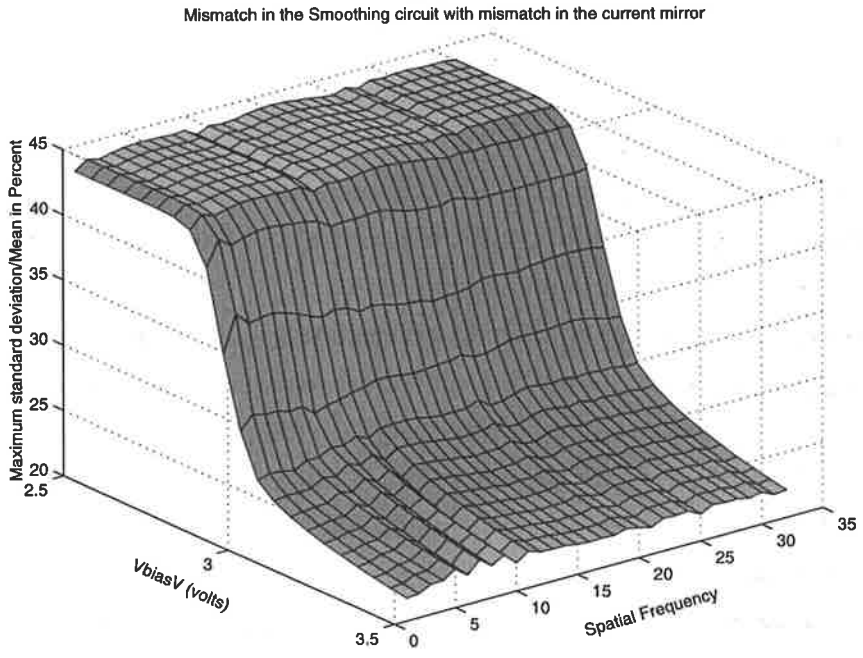
In Figures 6.7-e and 6.7-f, both circuits have been simulated with different biasing currents. I_{biasU} and $I_{BiasDiv}$ are set to 10 nA, instead of 100 pA for the previous simulations.

Here are the significant results derived from these simulations.

- Variance of the AB circuit is higher than that of the MNC circuit at higher spatial frequencies. However, when $vdown$ in the AB circuit is set around a practical value (from 1.4 to 1.6 volts in Figure 6.7-a), the variance is less than that of the MNC circuit.



(a)



(b)

Figure 6.5: Output variance in the smoothing circuit a) without mismatch, and b) with mismatch in the current mirrors. Notice the reversal of the direction of the $VbiasV$ axis (only for visualization purpose).

- The variance of the MNC circuit is relatively constant (notice the values on the Z axis). Also, from Figure 6.7-d, one can see that mismatch in the divider contributes to most of the variance of the output.

- One would expect to observe a higher variance at the output of the MNC circuit, due to the higher number of transistors, of about

$$\sqrt{\frac{13}{7}} \approx 1.35$$

  compared to the AB circuit. This is close to being the case if we compare the variance levels as seen in Figures 6.7-e, and 6.7-f, which is about $53/34 \approx 1.55$.

- Variance in both circuits decreases by increasing the biasing currents. In the subthreshold region, the σ_I/I term is almost constant, and hence the decrease of the variance can be associated with circuit nonlinearity. This is partly due the body effect in the transistors, as the substrate node of the transistors is not connected to the source.

In spite of being time consuming, simulations are necessary to identify the circuits which contribute most to the mismatch. For example, in the MNC circuit the divider accounts for about half of the mismatch.

Several conclusions with regards to minimizing the effect of mismatch are: to use larger transistors, to use the minimum number of transistors, to use larger currents, and to avoid subthreshold circuits. Unfortunately, these are in contrast with the requirements of vision chips, which are: low currents (subthreshold), small transistors, and the integration of as many transistors as possible (to increase functionality per pixel).

The amount of output variance in contrast enhancement circuits can be as high as 120%, which corresponds to the $(1\sigma)/mean$ value. Therefore, even for circuits with about 30% variation, one should expect single cell variations of about 100%. This high level of mismatch has been observed in many vision chips designed to date. Unlike simple imagers, in which the effect of mismatch can be alleviated by using correlated double sampling, using these these techniques in vision chips is almost impossible because the amount of output variance depends on many parameters, including the spatial content of the input and the mean input level. Analog VLSI systems of this kind may not be used in practical applications, unless adaptive mechanisms are developed to fix this problem.

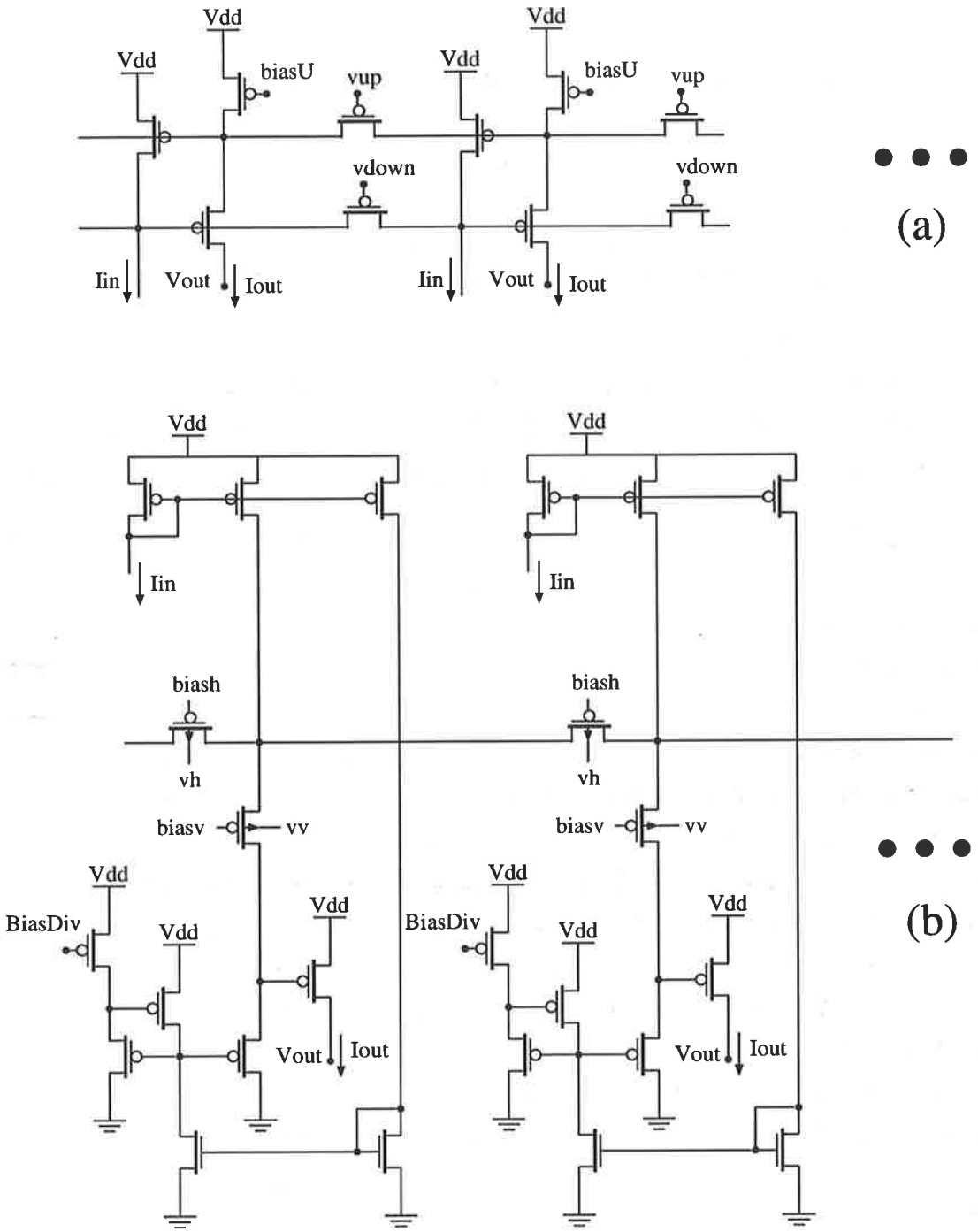
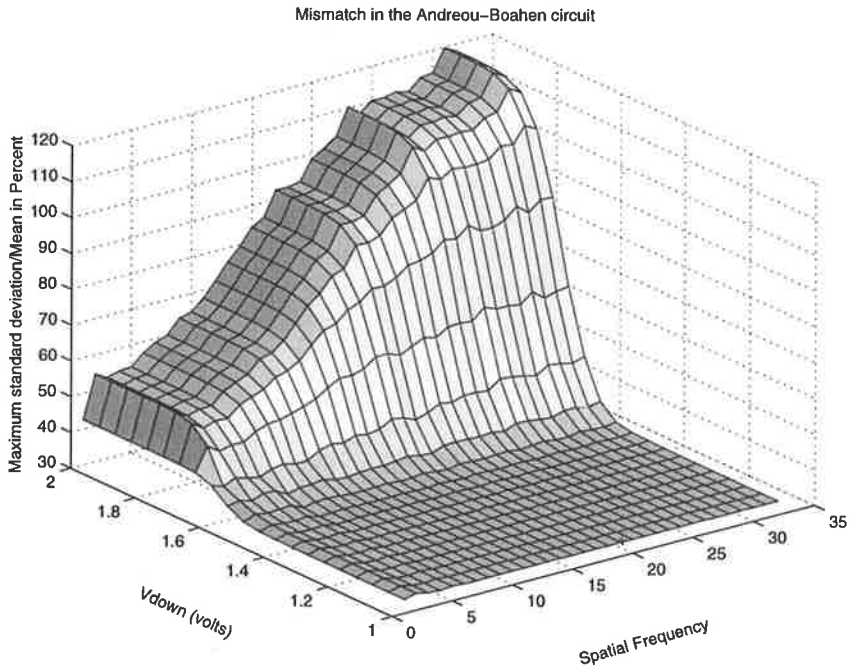
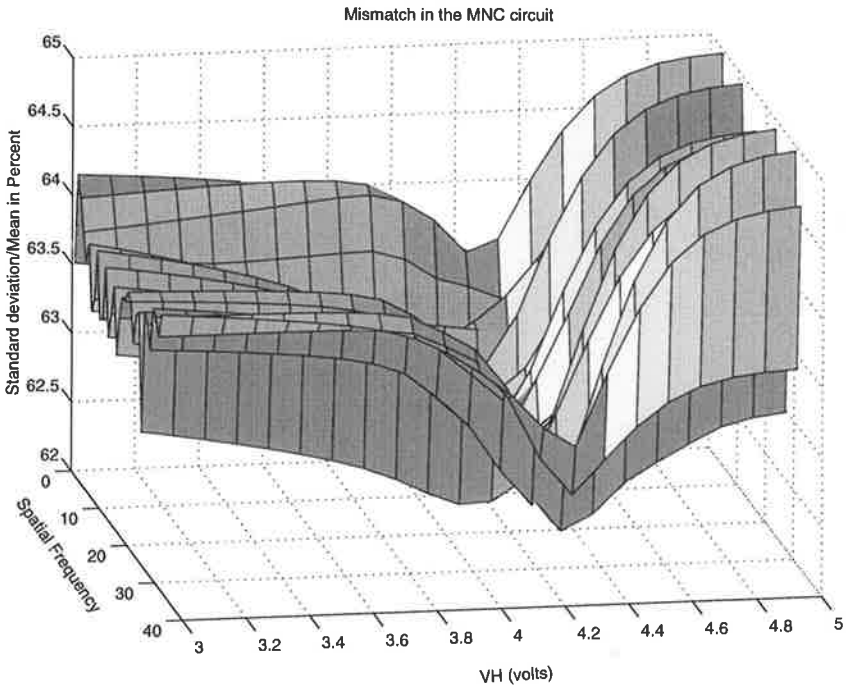


Figure 6.6: a) Andreou-Boahen's silicon retina. b) The MNC circuit.

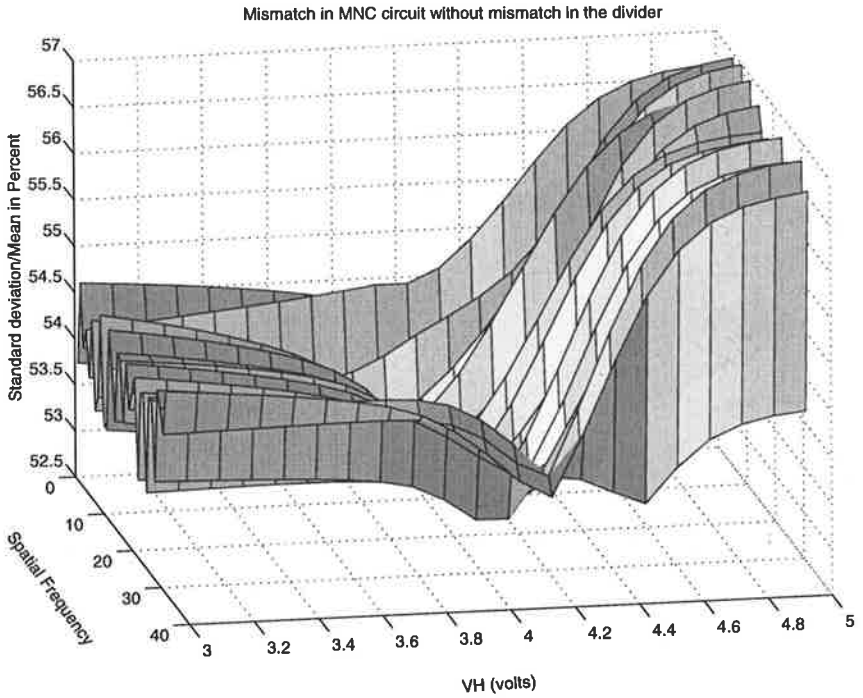


(a)

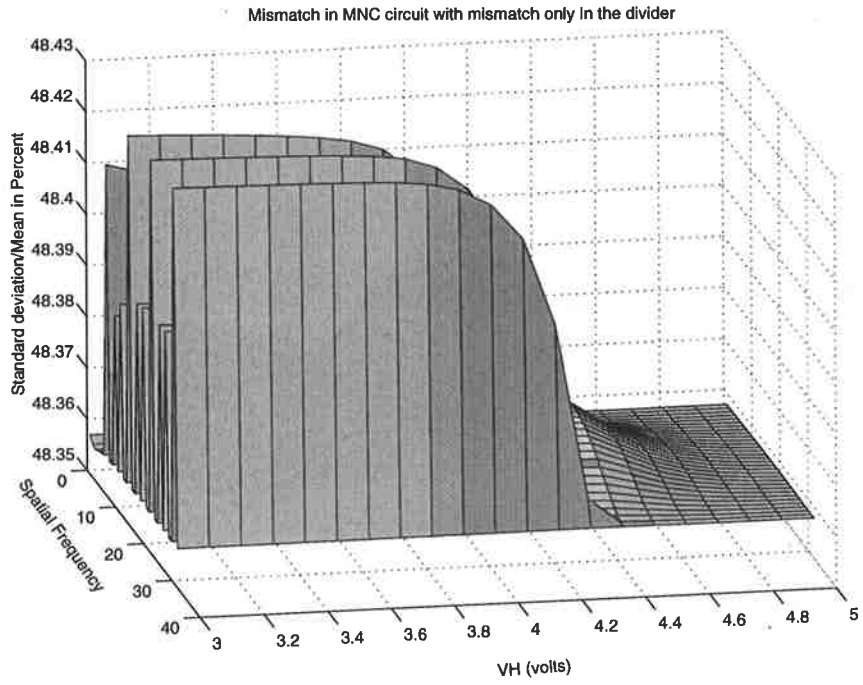


(b)

Figure 6.7: Output variance due to mismatch for the AB and MNC circuits.

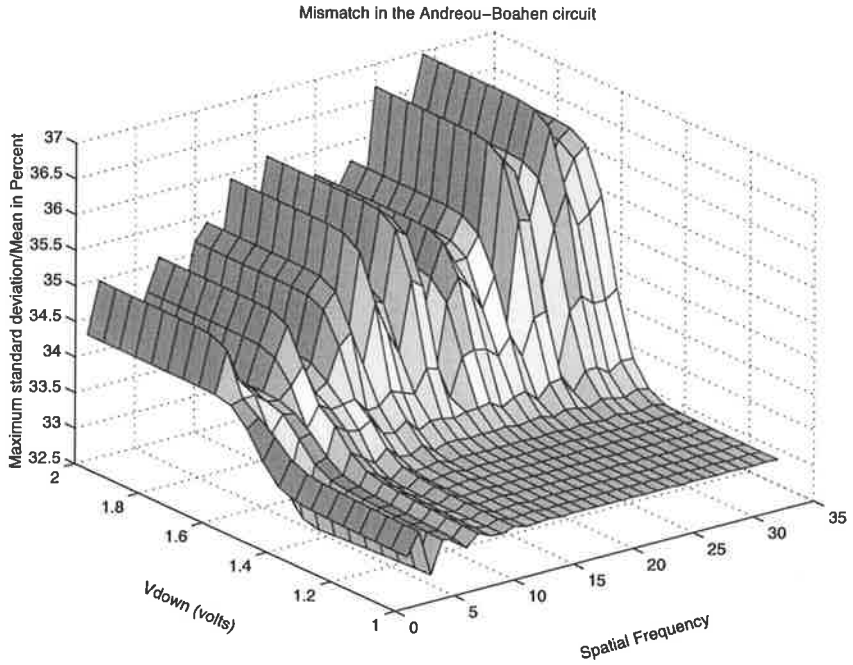


(c)

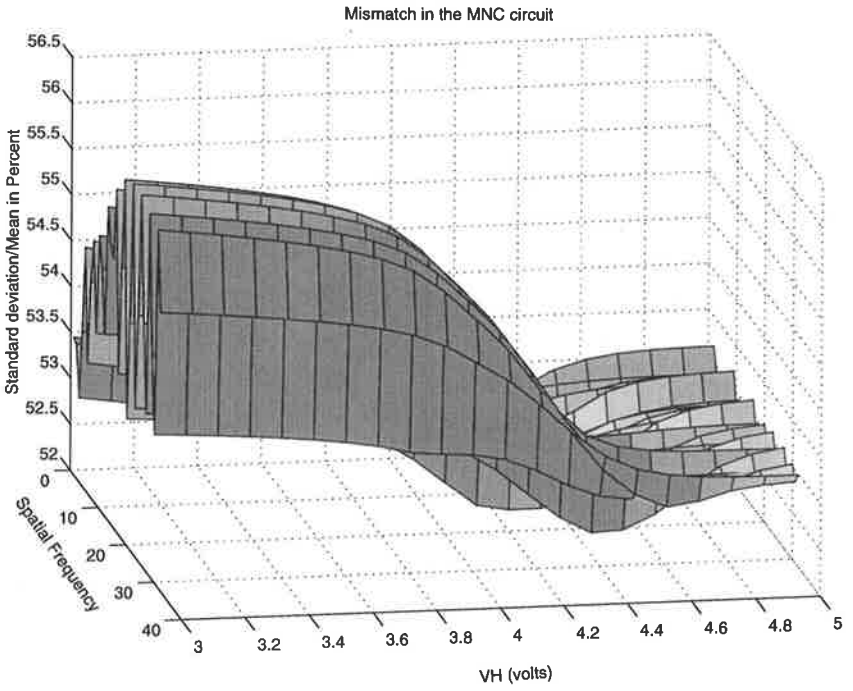


(d)

Figure 6.7: (*continued*).



(e)



(f)

Figure 6.7: (*continued*).

6.2 Interconnection

Interconnection is one of the limitations in achieving large AVLSI networks. Leading edge digital processes are concerned more about electrical characteristics (capacitance and resistance) of the interconnections. In AVLSI systems it is the density of interconnects which is more important.

Sometimes, an analogy is made between biological and VLSI systems by referring to their interconnect limitations. However, there is an important difference between the two, which is more prominent than the similarities. Biological systems use a 3-D interconnection structure, while current VLSI systems are limited to a 2-D planar structure. Even if the interconnection density inside a chip increases tenfold, the transfer of information off the chip remains limited to a maximum of about 500 signals for current technologies. Practical complexities arise from scanning a 2-D array into a 1-D stream of signals and then reconstructing the 2-D signal in the next processing level.

Therefore, one inspiration from nature would be to construct 3-D VLSI structures. This can be performed either at the chip level as suggested by [Kioi et al. 92], or at the packaging and board level [Crowley and Vardaman 94, Larcombe et al. 95, Stern et al. 96]. 3D multichip modules and chip-on-chip technologies are already in use for applications such as very high density memory modules. In these technologies the interconnects are still brought to the periphery of the chip. However, vision chips require a *chip level* 3D technology, which is still immature.

6.2.1 Interconnect density

The number of interconnects required between a cell and its neighbors in a vision chip is determined by the algorithm. As was discussed in Section 5.3 the number of interconnects for a four-sided cell is given by

$$M_1 = 2((2N+1)^2 - 1)$$

where N is the neighborhood involved in the processing of a cell. The additional area for the interconnects (only those to the neighbors, not the internal interconnects) is approximately

$$A = [S + N(N+1)P]^2 - [S]^2 \tag{6.11}$$

where S is the length of a side of the cell (in a square shaped cell), and P is the pitch of the interconnects. In most fabrication processes, the pitch of the metal interconnects is about four times the minimum feature size. If we rewrite the above equation in the units of λ, which is half of the minimum feature size, we have

$$A = [a\lambda + N(N+1)b\lambda]^2 - [a\lambda]^2 \tag{6.12}$$

Assuming that there are only N connections needed to the nearest neighbor, we will have

$$A = [a\lambda + Nb\lambda]^2 - [a\lambda]^2 \tag{6.13}$$

Figure 6.8 shows the percentage of area taken by the interconnects. In Figure 6.8-a, the additional area is excessively high, and therefore full connectivity between all neighbors is impractical.

One interesting point is that the use of a 3-D structure (at the chip level), does not significantly reduce the interconnect area. In a 3-D VLSI structure the cells and interconnects to the neighbors still remain in one plane, and only the output of the cells are transferred to the next processing plane. If a 2-D structure is used, the output bus and the select lines (which are often perpendicular to each other) can be considered as one extra connection to the neighbors. Therefore, the difference in the area of the cell between 3-D and 2-D structures will be the difference between two cells with N and $N + 1$ interconnects. For example, considering Figure 6.8-b, if the size of the cell is 100λ and there are 2 interconnects to the nearest neighbor, then the interconnect area for a 3-D structure will be about 35% and for a 2-D structure will be about 55% of the pixel area without the interconnects.

The main advantage in using 3-D structures is that there is no need for scanning circuits, the speed of information transfer is significantly increased, and there is little energy consumed for transferring the information.

6.3 Digital Noise

The effect of digital noise in mixed analog/digital designs has been studied and various design guidelines have been proposed [Masui 92, Su et al. 93, Makie-Fukuda et al. 95, Verghese et al. 96]. In most of these studies it is assumed that the digital and analog signals and modules are physically separated, and that digital noise is caused mainly by coupling through the power lines or the substrate.

The amount of noise coupling through the substrate appears to be dependent on the relative locations of the digital and analog circuits, and remains almost constant when the separation is above a certain distance [Verghese et al. 96].

In most AVLSI systems, including vision chips, digital and analog circuits are mixed, and the power, bias, read-out, and select lines are interwoven as show in Figure 6.9. This may not be a problem for circuits which only perform spatial processing operations, as the signals are allowed to settle to their final values within a time limit. However, any type of digital noise may be detrimental to the operation of spatio-temporal circuits, such as motion detectors, which are typically sensitive to temporal variations in the signals. For example, Figure 6.10 shows the output of a continuous time temporal contrast detection (TCD) circuit in the presence

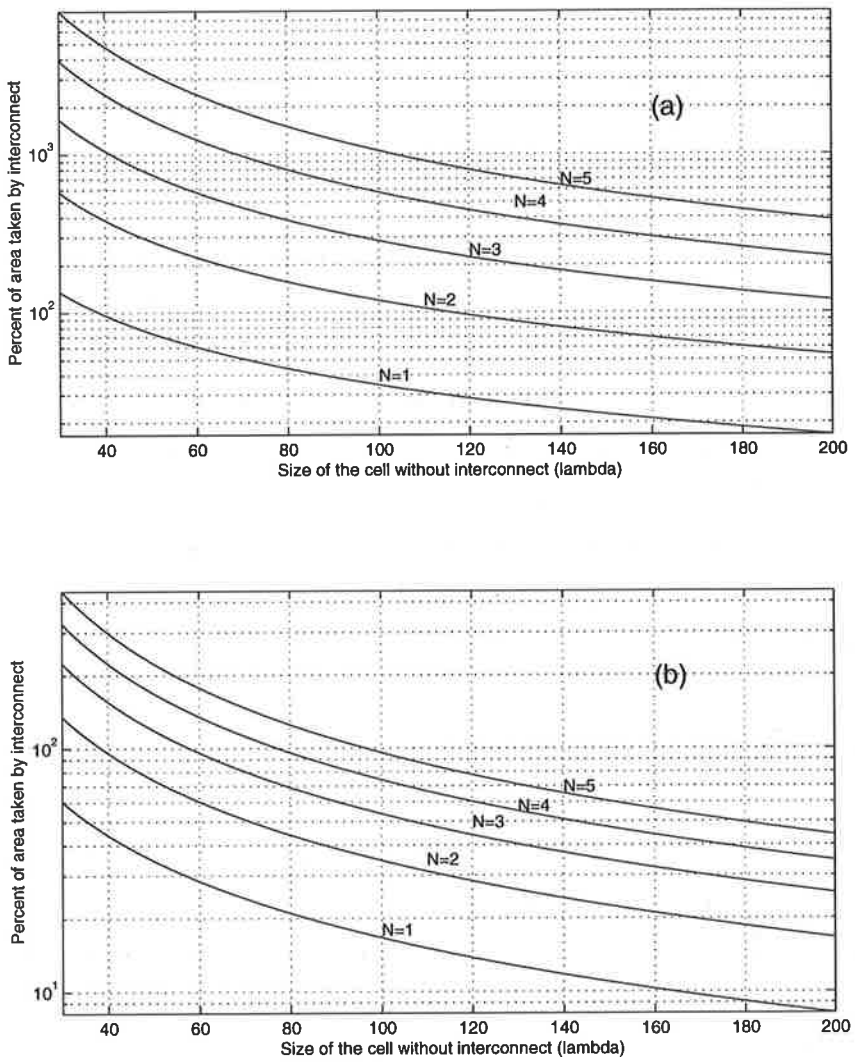


Figure 6.8: The percentage of area occupied by interconnections. a) With connections to all the cells in a neighborhood N. b) With only N connections emanating from each side.

and absence of digital noise, caused by scanning signals. The simulation results presented in Figure 6.11 also illustrate a similar effect as a result of 2 mV transitions on the supply voltage. The specific circuit was designed for high sensitivity to temporal variations at its input node.

Conventional methods for reducing digital noise are: increasing signal strength in the circuits by using higher currents, using low-swing and ramped switching voltages, using differential signals, reducing the resistivity of the power lines, and increasing the resistivity of substrate. Other technologies, such as silicon-on-insulator (SOI), can be used to isolate the substrates of the analog and digital circuits.

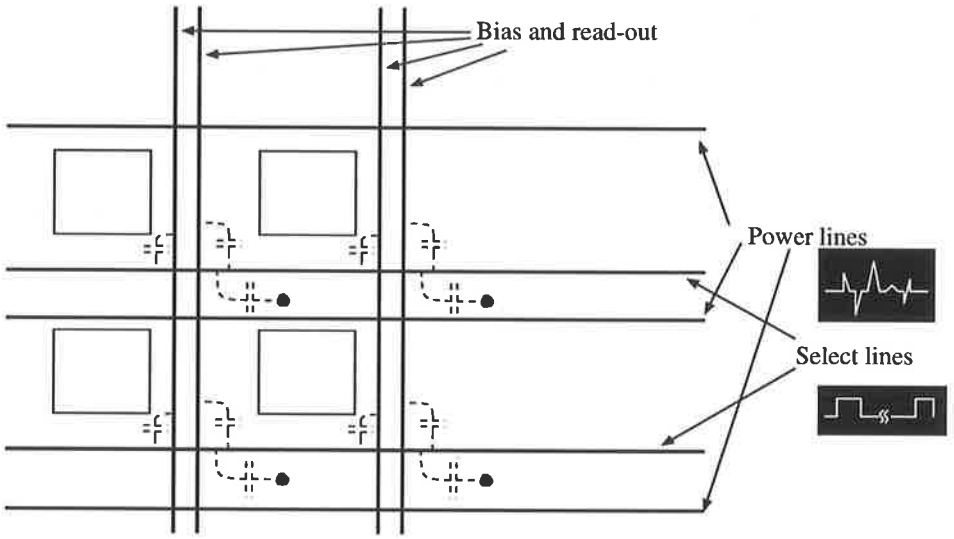


Figure 6.9: Sources of digital noise coupling in a vision chip.

6.4 Summary

This chapter provided insight into issues in the design of AVLSI systems. Mismatch as the most concerning problems in analog circuits was discussed. Its sources and its effect on circuits and networks for spatial processing were described in Section 6.1. It was shown that AVLSI systems may not be used in practical applications, unless adaptive mechanisms are developed to fix the mismatch problem.

Interconnection limitation, and its effect on the cell area and density were presented in Section 6.2.

Digital noise as another problem, which particularly affects spatio-temporal circuits and networks was addressed in Section 6.3.

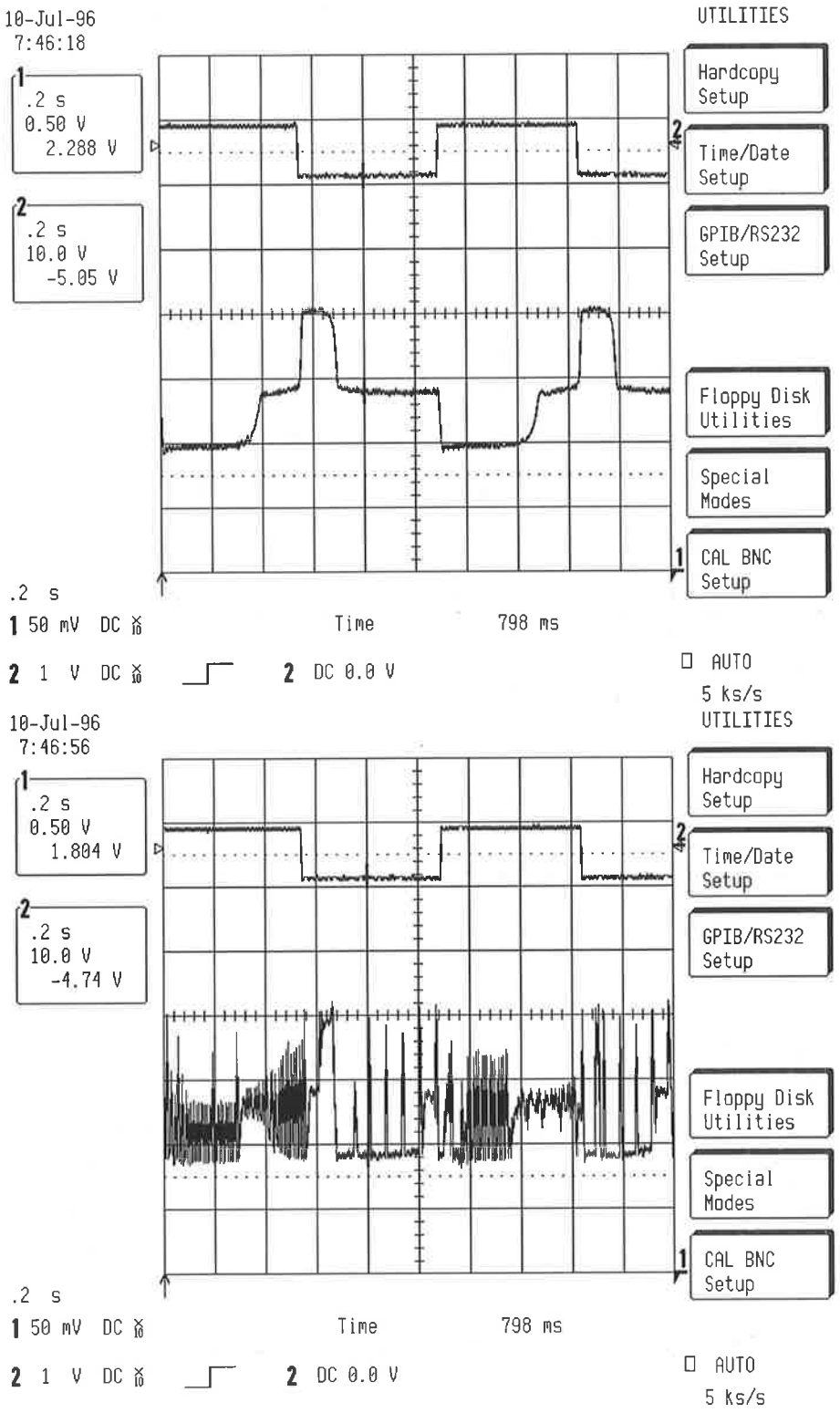


Figure 6.10: The output of a TCD circuit without and with digital activity in the chip.

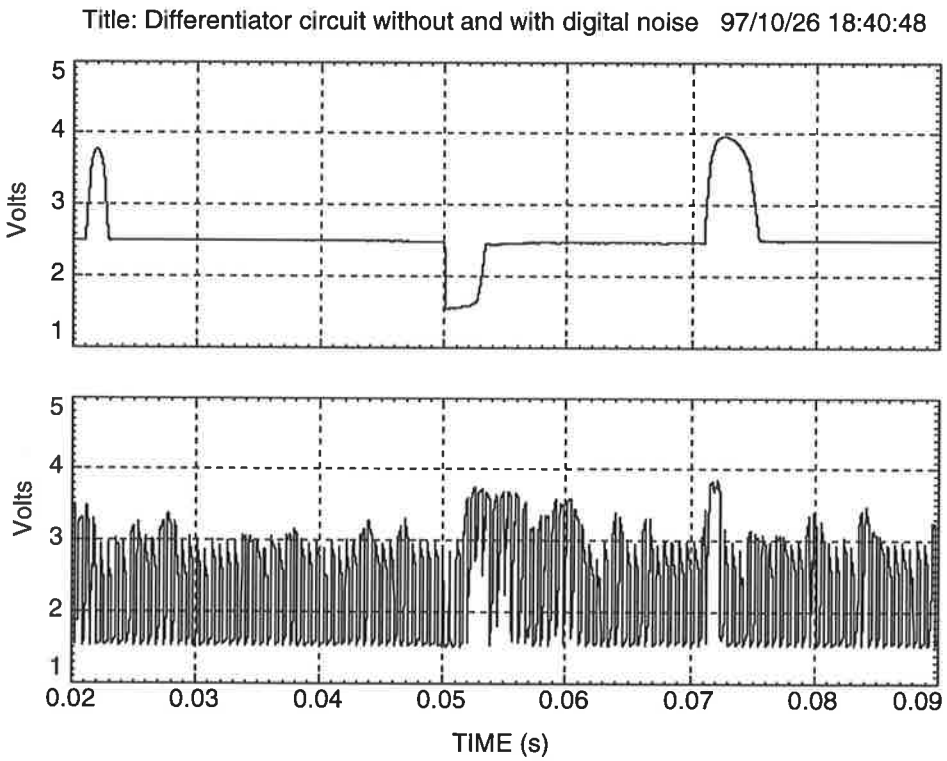


Figure 6.11: The effect of the supply noise on the output of a TCD circuit.

Part III

Designed Vision Sensors

Chapter 7

Introduction

During the course of this research several vision chips were designed and fabricated. The sequence and the purpose of the design of the chips were briefly described in Section 1.3. As summarized in Table 7.1 which lists the main features of the motion detection vision chips. Only Bugeye I and Bugeye V comprise all the components for local and global motion detection based on the template model. Bugeye III contains test structures for temporal contrast detection circuits. Bugeye II and Bugeye IV only comprise analog front-end circuits for temporal contrast detection.

Of these chips Bugeye II and Bugeye V will be described. These two chips address two important aspects of the design of vision chips, the circuit level and the architectural level. They also represent the first and the last steps in the development of a vision chip. Bugeye I, which was a proof-of-concept design has been described in detail in [Moini 94]. Chapters 8 and 9 describe the Bugeye II and Bugeye V chips, respectively.

Another chip, MNCSI (Multiplicative Noise Cancellation & Shunting Inhibition), contains various networks for contrast enhancement, and implements different versions of the shunting inhibition (SI) [Moini et al. 97a], and the multiplicative noise cancellation (MNC) circuit [Moini and Bouzerdoum 97]. MNCSI chip is described in Chapter 10.

Table 7.1: Features of the Bugeye chips.

Characteristic	Bugeye I	Bugeye II	Bugeye III	Bugeye IV	Bugeye V
Purpose of design	Proof of concept	Algorithm and circuit development	Test structure for TCD circuits	Smart imaging for algorithm development	Implementing a 2-D motion detector
Algorithm implemented	Template Model	Template Model	None	None	Template Model
Number of channels	1×64	2× 64	34 circuits	32×64	4×64
Photodetector structure	Single mode photodiode	Double sensitivity phototransistor	Double sensitivity phototransistor	Double sensitivity phototransistor	Phototransistor
Analog processing performed at pixel level	Logarithmic AGC & Differentiation	Logarithmic AGC, MNC & Differentiation	Temporal contrast detection	Logarithmic AGC & Integration	Logarithmic AGC, Differentiation, and Thresholding
Other features	Analog & digital functions on the same chip	High flexibility and testability	–	Parallel A/D conversion	–
Process Used	2μ 2P-2M CMOS	1.2μ 2P-2M CMOS	0.8μ 1P-3M CMOS	0.8μ 1P-3M CMOS	2μ 2P-2M CMOS
Publications	[Moini et al. 93] [Bouzerdoum et al. 94] [Nguyen et al. 93] [Yakovleff et al. 93] [Nguyen et al. 94] [Moini 94]	[Moini et al. 97b] [Moini et al. 95a] [Moini et al. 95b]	[Moini et al. 96]	[Moini et al. 96]	[Moini and Bouzerdoum 97]
Area (mm^2)	4.6×4.5	2.2×2.2	1.9×1.9	2.7×4.4	1.6×3.2

Chapter 8

Bugeye II: The Second Insect Vision Motion Detection Chip

Test results from Bugeye I revealed several problems in various parts of the design. Firstly, low-level testability was not considered in the design of Bugeye I because it was only meant to be a concept demonstration. There was no direct access to the analog front-end, and the circuits were hard-wired, which meant that their function could not be changed. Secondly, very small biasing currents, in the order of several pico-amperes, were required for the temporal contrast detector (TCD) circuits. Thirdly, the chip could not function properly under AC light sources, whose fluctuations would be detected by the TCD circuits.

Following an overview of the architecture of Bugeye II, this chapter addresses these issues and describes the implementation of the analog front-end.

Bugeye II implements the first processing stages of the template model, i.e. photodetection, temporal contrast detection, and thresholding (see Figure 2.3). The chip was designed mainly as a test-bed for analog circuits, therefore template formation and motion detection are performed off-chip.

Bugeye II comprises two one-dimensional arrays, each comprising 64 elements. The only difference between the two arrays is the placement of special analog switches, which are used for configuration and testability purposes, as described in Section 8.1. Figure 8.1 shows the architecture of Bugeye II. A photodetector generates a photocurrents, which is processed by a MNC circuit, in order to reduce the AC noise. Note that MNC is not a part of the template model. The temporal contrast detection unit (differentiator) detects temporal changes in its input current. The change is then thresholded, yielding two bits which are sampled and stored using digital latches. Row selection is performed by using a shifter and a multiplexer. The bidirectional analog switches can be configured in such a way that the output of a cell can be read out or a signal can be applied to the cell.

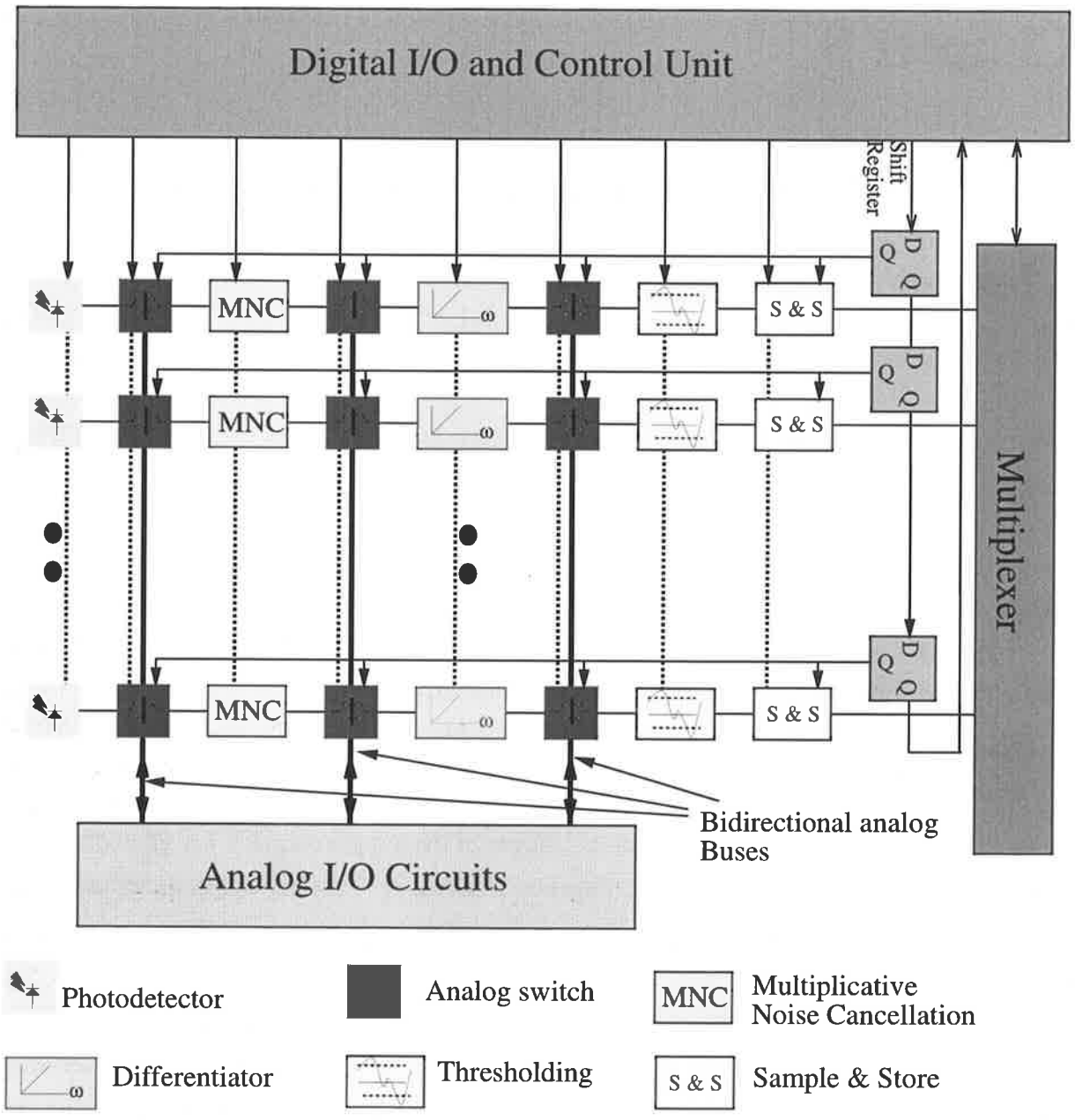


Figure 8.1: The architecture of Bugeye II.

8.1 Testability and Reconfigurability

Testability

In order to be able to test each cell in the array, an analog test bus is used in conjunction with analog switches, as illustrated in Figure 8.2. A cell is selected by activating its *Select* signal, and the *COMM* line, which is common to all the cells in one column, is used to transmit the signals to the pads. By using the two control signals *Cin* and *Cout*, either the output of the *Previous circuit* or the input of the *Next circuit* can be connected to the *COMM* line. When *Select* is not activated, the switches are connected to the *Internal* signal, and do not interact with the *COMM* line.

The parasitic capacitance associated with the long *COMM* wire and the pad, and the external capacitance of measuring equipment, will obviously affect the transient response of the circuits, and hence the signals should be buffered before being sent to the pad. However, we have chosen not to do so, due to the nonlinear behavior of most available buffers, and also to the fact that only the DC current levels are of interest. Therefore, only an analog switch at the pad has been used.

Reconfigurability

In the early stages of system development many unknown factors may affect the function and behavior of the system. Therefore, some degree of flexibility should be planned. In analog arrays containing several new circuits, it is preferable to design the circuits in such a way that they can be selected from several different alternatives. This can be achieved by implementing the different alternatives and selecting the desired one. This approach, however, requires a large area.

In some cases, the topology of the circuits may suggest a simpler method than implementing all alternative circuits and switching between different configurations. In the following the reconfigurable circuits that were implemented in Bugeye II are introduced. More detailed descriptions of the function of each circuit are provided in succeeding sections.

The photodetector has been implemented with a vertical bipolar transistor between substrate, well, and diffusion. Hence two alternatives are available: the collector-base junction diode by connecting the base and emitter together, or the bipolar transistor by leaving the base floating (see Section 8.2).

The size of the averaging window of the current-mode spatial averaging circuit (CMSA), used in the MNC operation, can be selected from three different values. The divider used in the MNC operation can also be switched on or off [Moini et al. 95a], resulting in different functionalities (see Section 8.3 for detailed descriptions).

The topology of the resistive element used in the differentiator can be selected from a simple

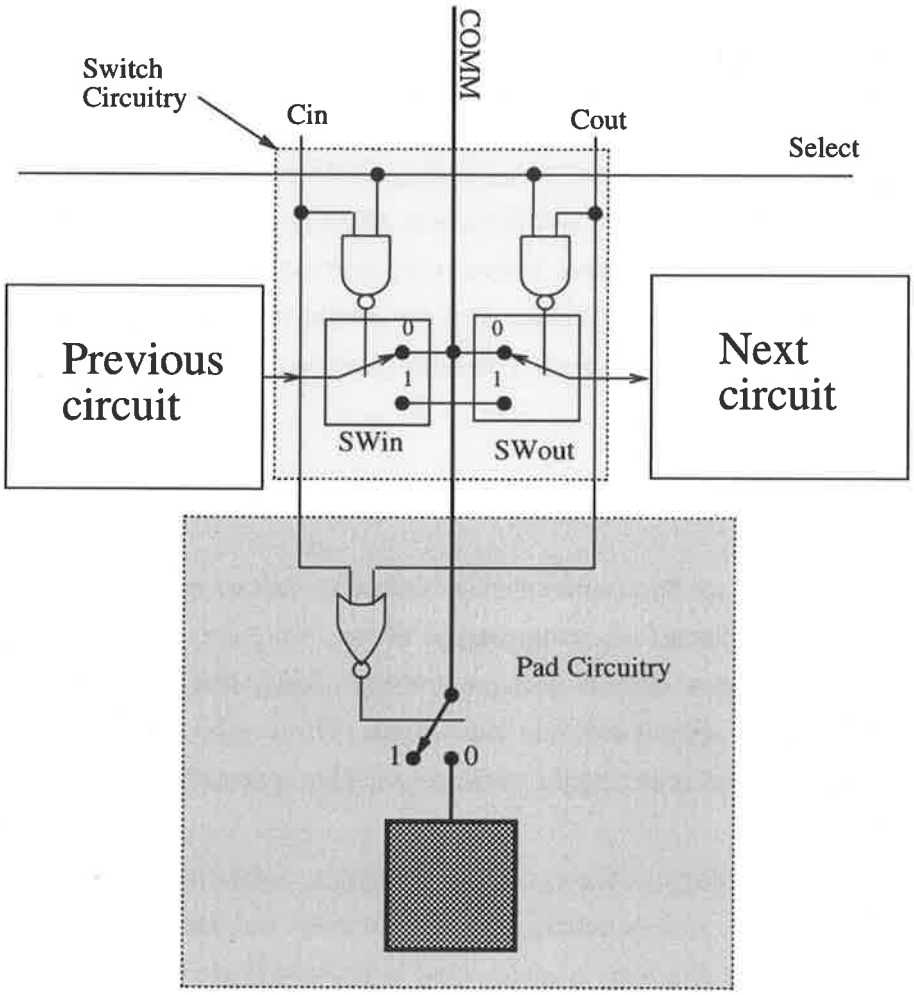


Figure 8.2: Circuit diagram of the switching circuit.

OTA circuit or a channel-length modulation-based OTA (CLM-OTA); the details are presented in Section 8.4.

The switching element used for changing the topology of all of these circuits is a single MOS transistor. Therefore, for these circuits configurability comes at a very low cost in terms of area.

8.2 Photodetector

The photocircuit used in Bugeye II is based on the logarithmic circuit, described in Section 5.2.1, and uses two MOS diodes, as shown in Figure 8.3-a. The photodetector is a vertical parasitic bipolar transistor between the N-substrate/P-well/N-diffusion. Transistor M1 switches the photodetection mode to either the photodiode, or the photobipolar mode. The test results of Figure 8.3-b show the output current of the detector in the two different modes. The current gain of the photobipolar compared with the photodiode is about 120.

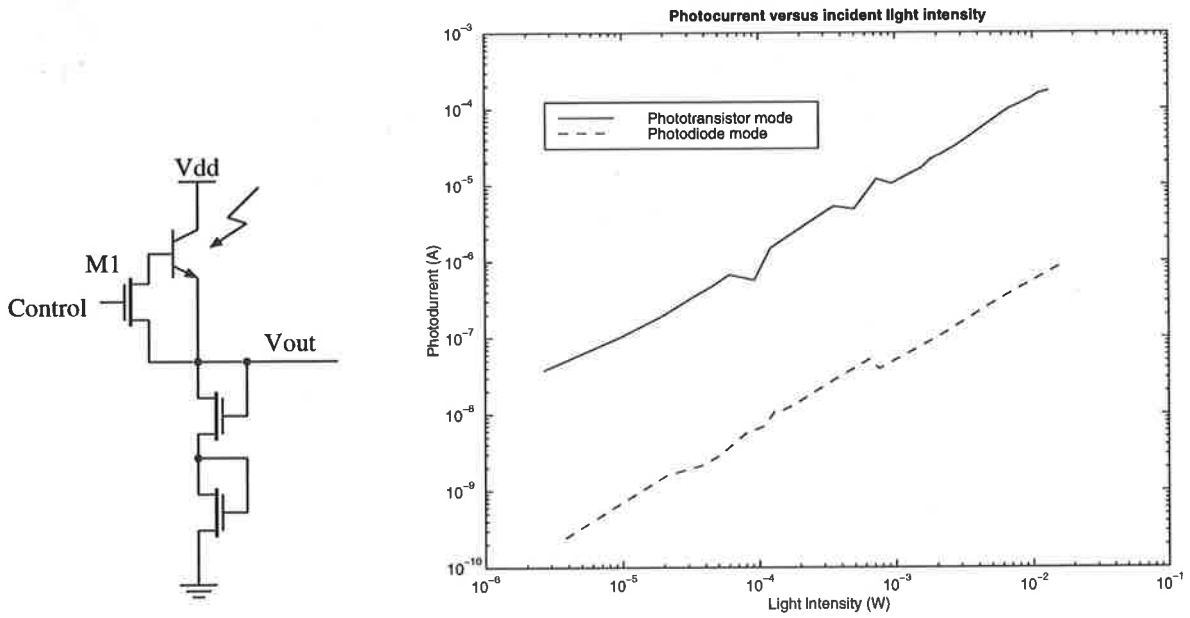


Figure 8.3: a) The photocircuit used in Bugeye"II. b) The output current versus input light intensity.

8.3 Multiplicative Noise Cancellation

The source of multiplicative noise

The reflectance of a light source from the surface of an object depends on the geometry and optical characteristics of the object. For an opaque object with a *Lambertian*, surface the relationship between these parameters can be described by

$$L = \frac{\rho(\lambda)}{\pi} E \cos \Theta_i \tag{8.1}$$

where L is the luminance of the object, ρ is a wavelength dependent reflectance coefficient, E is the illuminance (the amount of incident light), and Θ_i is the incidence angle. For multiple reflections the situation becomes more complex, but this equation can be considered as a first order approximation.

Light sources operating on mains power supplies carry a strong 100Hz component (or 120Hz in some countries). The amplitude of the detected AC noise can easily exceed that of the signal, when differentiated. In the following discussions the noise referred to is the AC noise, not other types of noise.

In additive noise, the frequency characteristics of noise and signal are added together. If the signal and noise are separated in the frequency domain, additive noise may be cancelled. In the case of multiplicative noise, however, the frequency characteristics of the noisy signal is a convolution of the frequency responses of the signal and the noise. Therefore, it is almost impossible to separate the noise and signal in the frequency domain, unless the noise has a single frequency component. In this case, which is the case for AC noise, the frequency response of the noisy signal will be a shifted version of the frequency response of the signal without noise.

Reducing multiplicative noise

There are several methods to cancel or reduce multiplicative noise. The first, called "synchronous sampling", consists of sampling the noisy signal at the same rate as the noise frequency. Therefore, if the amplitude of noise is constant, the sampled signal will only have a constant scaling factor. This approach limits the bandwidth of the input signal to half of the frequency of the noise, to satisfy the Nyquist criteria. The corresponding sampling rate may not be sufficient for applications where images should be sampled and processed at higher rates.

The second method uses filtering to reduce the noise. If the incident light has a DC level, then

$$L = \frac{\rho(\lambda)}{\pi} (E_{AC} + E_{DC}) \cos \Theta_i \tag{8.2}$$

where E_{AC} and E_{DC} are the AC and DC components of the light intensity. If the DC component is larger than the AC component, then in the frequency domain the effect of the AC

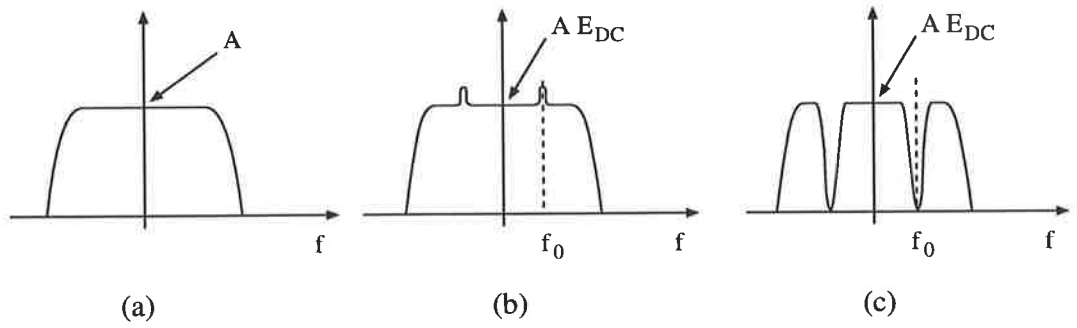


Figure 8.4: Reducing the AC noise by filtering. a) The spectrum of the input signal without noise. b) The spectrum of the signal and noise. c) The spectrum of the signal and noise after filtering. Note that it is assumed that the DC component of the light intensity is much larger than its AC component.

noise may be reduced by filtering the signal around the frequency of the noise, f_0 (see Figure 8.4). However, this will remove part of the frequency spectrum of the signal. Also implementing this filter is difficult in AVLSI. This method has been used in an implementation by Sarpeshkar et al. [Sarpeshkar et al. 96a] .

The third method, is based on collective signal processing, as distinct from single channel signal processing. It is reasonable to assume that in an array of pixels, the AC noise is relatively uniform over a local neighborhood. Therefore, if the signal in one channel is divided by the local spatial average of the signal, the noise will be effectively reduced [Moini et al. 95a, Moini et al. 97b]. Figure 8.5 illustrates the implementation of this method, which was introduced in Section 2.3.1 in the context of contrast enhancement.

In order to implement the MNC operation using this approach, a spatial smoothing network and a divider are required. The spatial smoothing circuit shown in Figure 8.6 is used in Bug-eye II. The operation of this circuit was described in Section 5.3.3 (see Figure 5.18). The divider, shown in the shaded box in Figure 8.7, is a translinear circuit introduced by [Andreou et al. 91].

Figure 8.8 shows the simulation result for an array of 20 MNC cells. The input to the array is a spatial step function with current levels of $1nA$ and $2nA$ at each side of the step function (see Figure 8.8-b). Multiplicative noise with a frequency of $50Hz$ and a magnitude of 50% of the signal level in each channel has been added. The relative noise in the output signal has been significantly reduced from the initial 50% to less than 5% of the DC level of the signal. The output also clearly demonstrates the effect of contrast enhancement.

The MNC circuit was tested by projecting a light strip onto the array. Figure 8.9-b shows a comparison between the measured and simulated output when S1=S0=1. In the simulation, a simple rectangular averaging on a window size of three was performed on the "measured input".

Figure 8.10 presents the measured output of the MNC circuit, for the same input signal shown

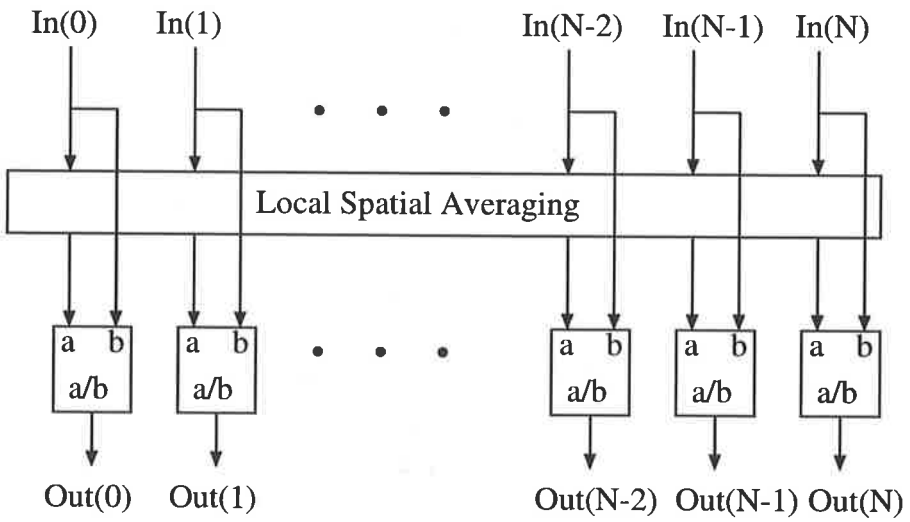


Figure 8.5: Cancelling the multiplicative noise using "division by spatial average".

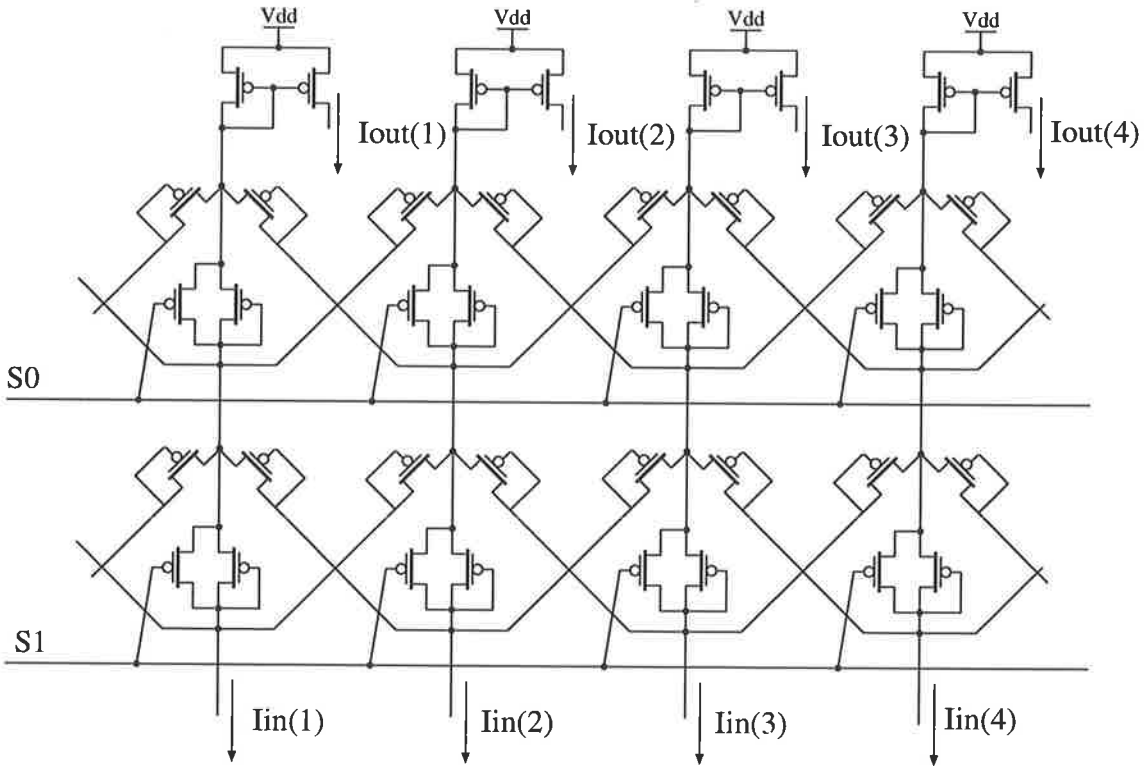
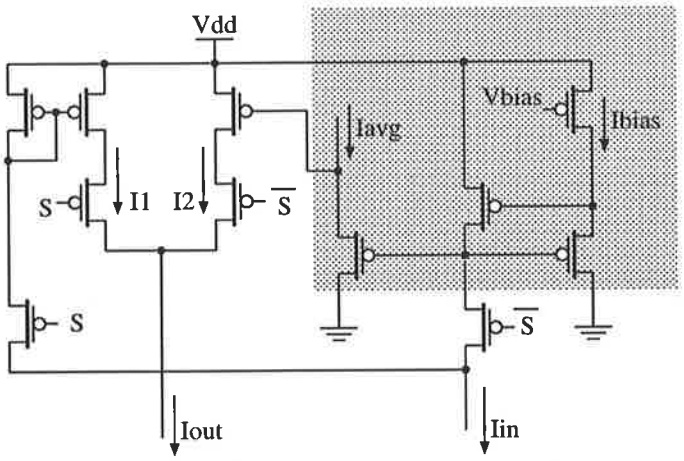


Figure 8.6: Spatial smoothing circuit implemented in Bugeye II.



if $S = 1$ $\quad I_{out} = I_2 = I_{bias}\frac{I_{in}}{I_{avg}}$

if $S = 0$ $\quad I_{out} = I_1 = I_{in}$

Figure 8.7: The translinear divider/multiplier used for MNC. I_avg is the output of the spatial smoothing circuit.

in Figure 8.9. In this test the smoothing circuit is configured so that both stages of the spatial smoothing are activated (S1=S0=1). The edges in the input are preserved in the output, while the spatial DC levels at the output are compressed. The fluctuations observed in the measured signals are due to mismatch in the circuits.

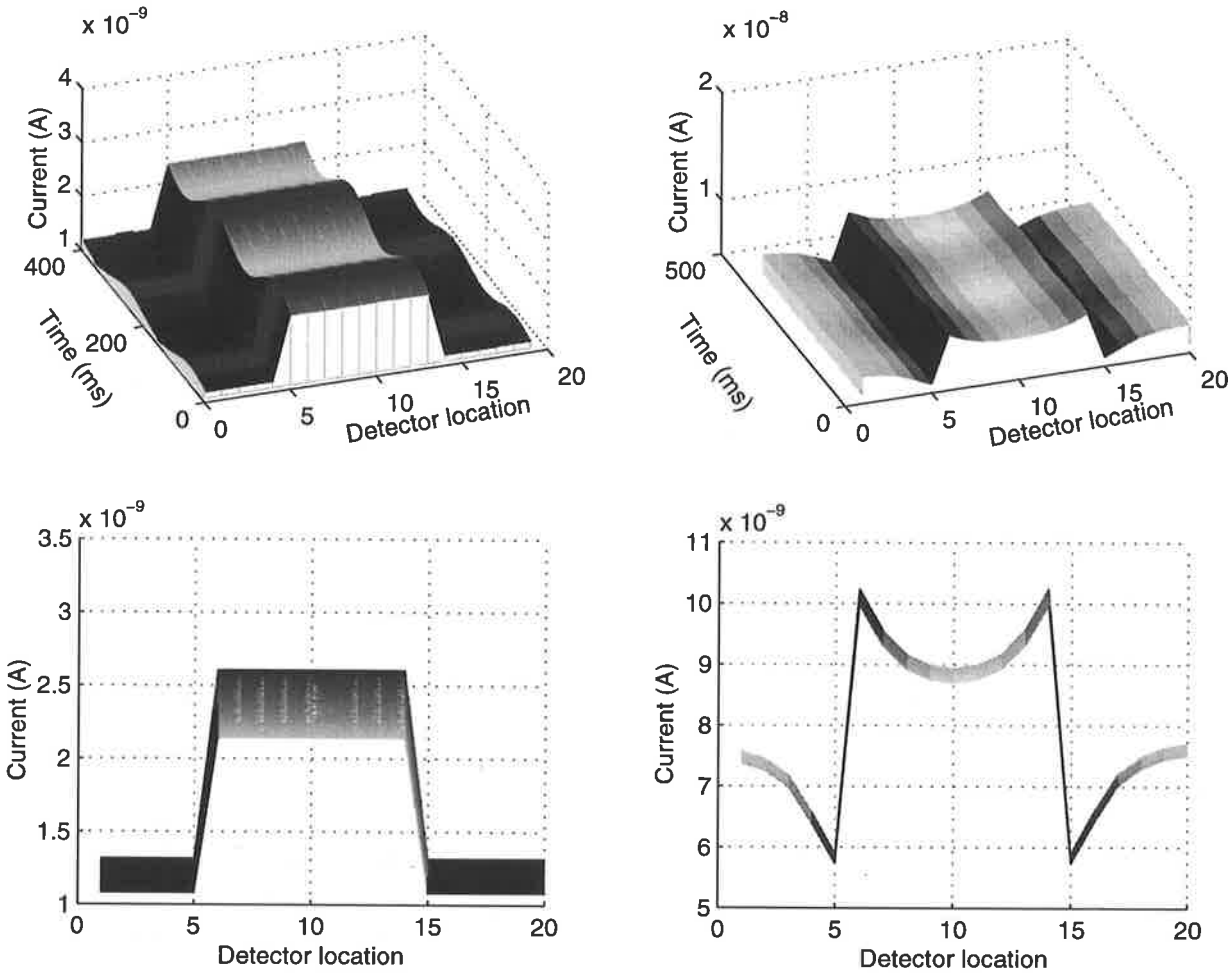


Figure 8.8: Simulation result of the MNC circuit. The top-left plot is the input, and the top-right is the output. The bottom plots show a different view of the input and output signals.

8.4 TCD circuit

An important operation in the template model for motion detection is temporal differentiation. In the implementation of a temporal contrast detector (TCD), the bandwidth of the signals (in our case the photocurrents) to be differentiated is of major concern. Differentiator circuits implemented in VLSI have minimum and maximum frequencies for proper operation. These frequencies are determined by the gain of the differentiator, the parasitic capacitance and impedances present in the circuit, and the dynamic range and saturation voltage of the circuits used in the realization of the differentiator.

The OTA based circuit, shown in Figure 8.11, was used in Bugeye I [Moini et al. 93]. In order to yield a specified equivalent resistance, R_{eq}, the OTA in this circuit requires a biasing

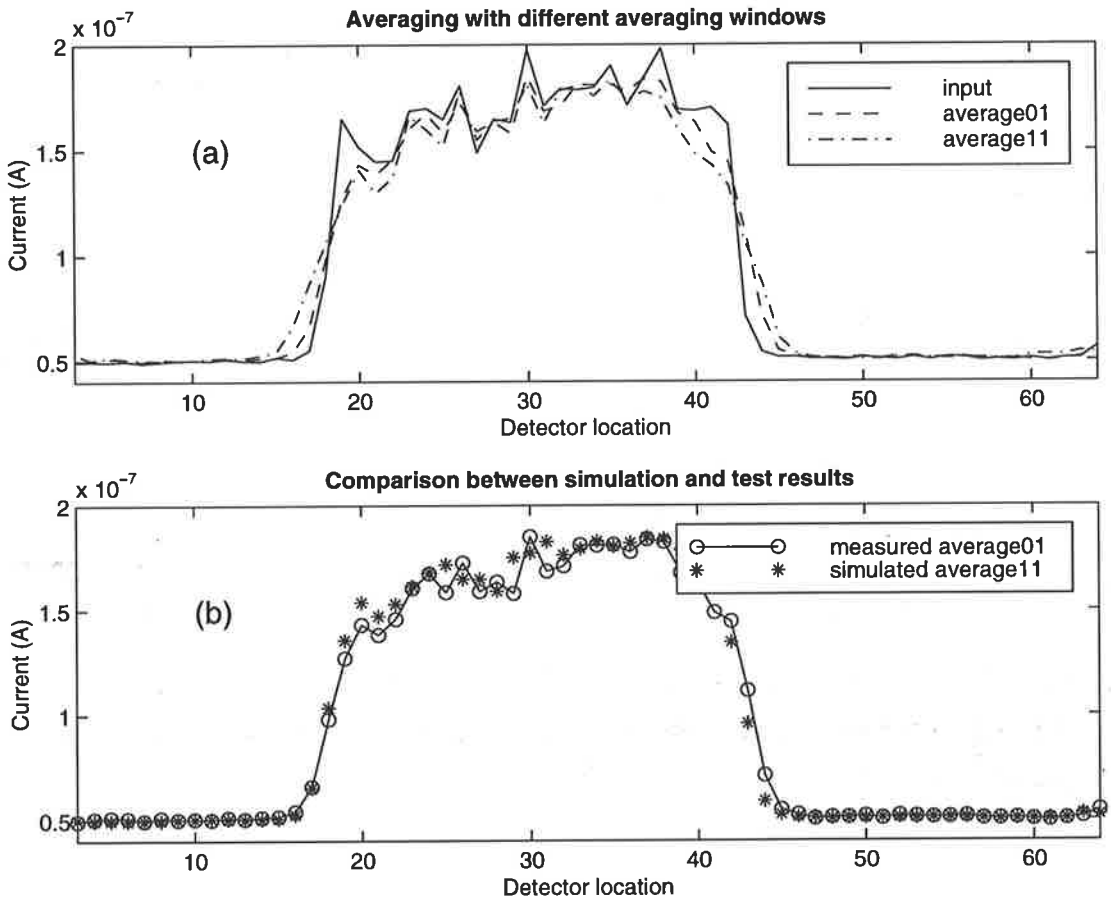


Figure 8.9: Test results of the spatial smoothing circuit in Bugeye II. a) The input, and the measured output of the smoothing circuit with two different windows. "average01" is the output of the when S1=0 and S0=1, and "average11" is the output when S1=S0=1. b) The measured and simulated output of the circuit.

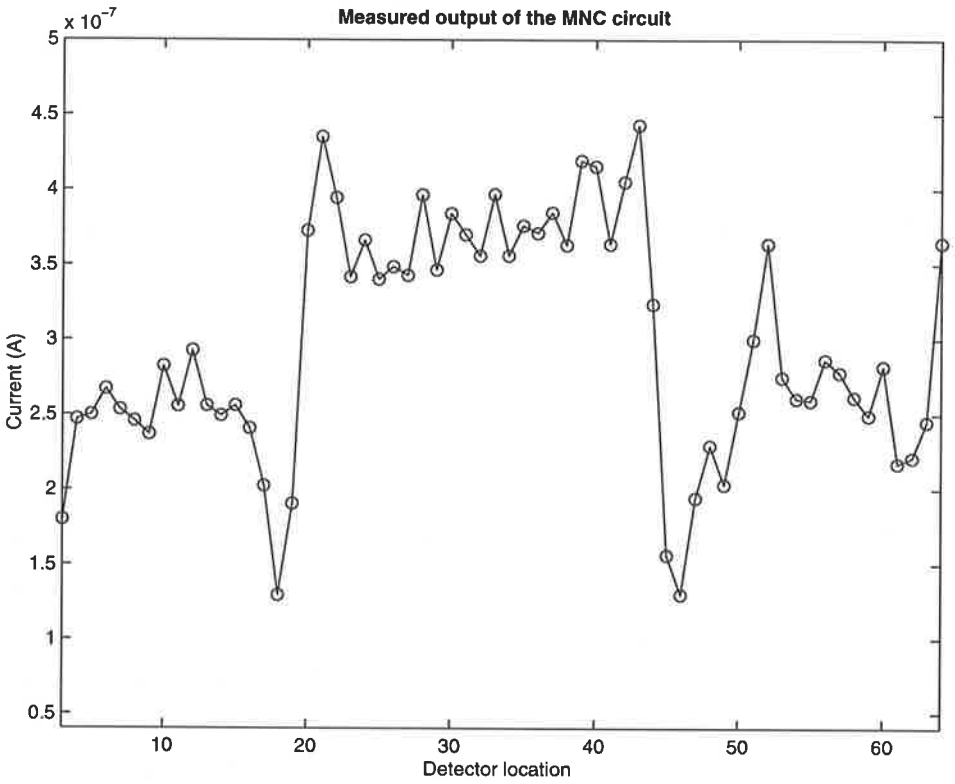


Figure 8.10: Measured output of the MNC circuit.

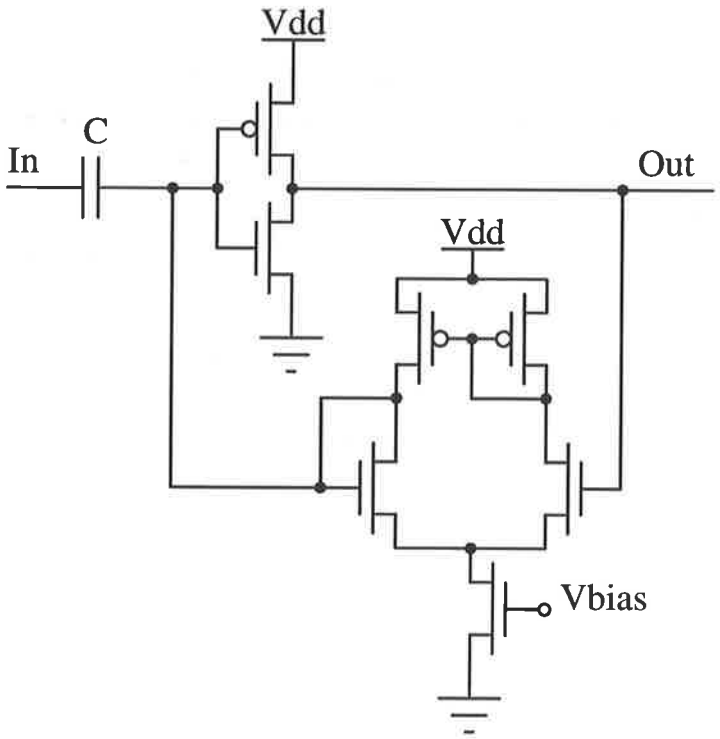


Figure 8.11: The OTA-based differentiator implemented in the Bugeye I.

current of

$$I_{bias} = \frac{kT}{nqR_{eq}} \tag{8.3}$$

where n is the subthreshold ideality factor. Very small currents, in the deep subthreshold region, are needed to yield the desired equivalent resistances, which are in the order of 1G Ohm.

In order to obtain such high resistance values while maintaining feasible biasing currents, an OTA based on the channel length modulation (CLM-OTA) in MOS transistors was designed. The simplest expression characterizing the drain-source current I_{ds} as a function of the channel length modulation coefficient λ is given by:

$$I_{ds} = f(V_{gs}, V_T)(1 + \lambda V_{ds}) \tag{8.4}$$

Figure 8.12 shows a simple circuit based on this idea. I_{bias} is the bias current. Transistors M2 and M1 form a source follower stage. V_b follows the input voltage V_{in}, and modulates the current through the input branch, I_1. Transistors M6 and M5 provide the reference current, I_2, which is subtracted from the modulated current using the current mirror formed by M3-M4.

By changing the state of M7, the circuit can operate either in the channel length modulation mode, when M7 is turned off, or in the simple OTA mode, when M7 is turned on.

Note that when the input voltage is very low, the current in the input branch is cut off, and the output current will be equal to the reference current $I2$.

The operation of the CLM-OTA can be affected by device mismatch, which stems from two main sources: the mismatch between the λ factors of the two transistors M1 and M5, and the mismatch between the current mirror transistors M3 and M4. In an ideal circuit, the output current would be zero, if the input voltage was equal to the reference voltage V_{ref}. For the purposes of analysis, it is convenient to assign all the fluctuations of β (the transconductance of transistors) and V_T (the threshold voltage) to the current mirror. This results in an offset in the bias current as a shift of the $I_{ds} - V_{ds}$ curve shown in Figure 8.13-a. The fluctuations of V_A (the Early voltage) of M1 and M5 appear as a change of the slope of the $I_{ds} - V_{ds}$ curve, as indicated in Figure 8.13-b. The total offset voltage ΔV_{offset} can be expressed as

$$\begin{aligned} \Delta V_{offset} &= \sqrt{(\Delta V_1)^2 + (\Delta V_2)^2} \\ &= \sqrt{\left(\frac{\Delta I_{bias}}{I_{bias}} V_A\right)^2 + \left(\frac{\Delta V_A}{V_A} V_{ref}\right)^2} \end{aligned} \tag{8.5}$$

where I_{bias} is the bias current, and V_{ref} is the reference voltage. $\Delta I_{bias}/I_{bias}$ and $\Delta V_A/V_A$ are the relative variances of the bias current and Early voltage due to transistor mismatch, respectively. $\Delta I_{bias}/I_{bias}$ is inversely proportional to the transistor gate length, L. V_A is directly proportional to L, and $\Delta V_A/V_A$ is relatively constant. The factor V_A in the first term of equation 8.5 may significantly increase the offset due to bias current mismatch. Hence, V_A should be chosen as small as possible, which means that the transistor gate length should be proportionally small. Also, the width of transistors M1 and M5 should be made large enough to decrease

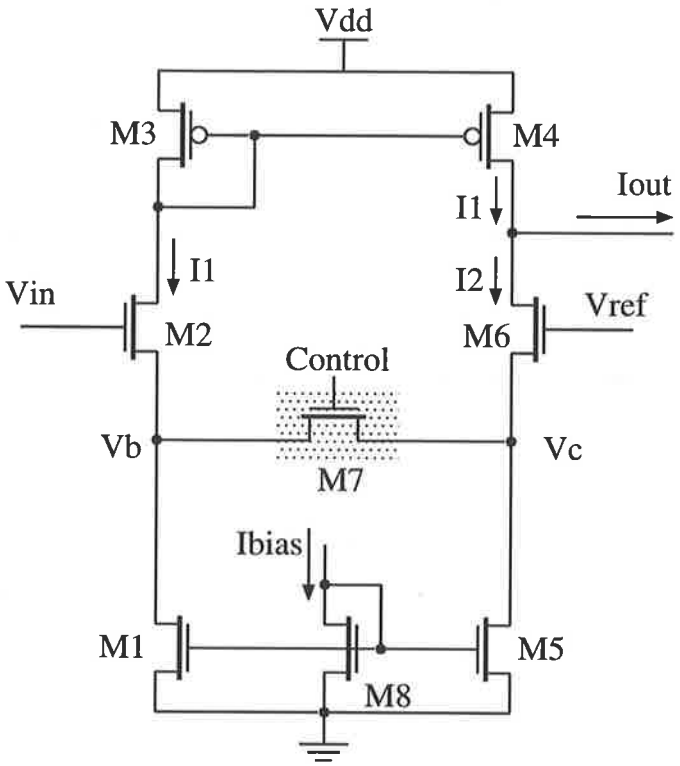


Figure 8.12: The reconfigurable transconductance amplifier, which can switch between the simple OTA-mode and the CLM-OTA-mode.

mismatch. In our design the minimum gate length, which is 1.2 μm, has been used for biasing transistors, whose Early voltage is about 5 volts.

In order to test the CLM-OTA, the reference input, V_{ref}, was fixed at 2.5 volts, while the other input was varied from the ground voltage to the supply voltage. The output current of the CLM-OTA is very linear over a range of 1 V to 5 V, i.e. from about the threshold voltage of the transistor to the supply voltage. Figure 8.14 shows the I-V characteristics of the OTA in the simple OTA mode and in the CLM-OTA mode.

In Bugeye II, an individual OTA circuit was implemented for testing purposes, as the The OTAs used in the TCD circuits in the array are in a feedback loop, and are not accessible.

The I-V characteristics of all the available CLM-OTAs were measured in all forty chips received from the manufacturer. At a biasing current of 100 nA, a mean offset voltage of 0.37 V with a standard deviation of 0.55 V is obtained. The maximum offset measured is about 1.2 V. The CLM-OTA is used in a feedback loop in the TCD circuit, and the DC offset of the circuit does not affect the operation of the next processing stages. Therefore, large offset values do not affect the function of the TCD circuit.

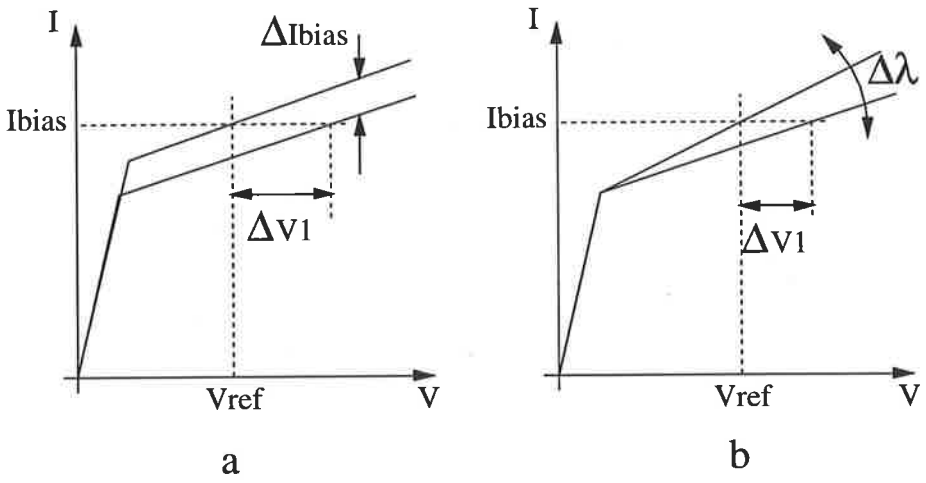


Figure 8.13: Offset voltage introduced by mismatch between transistors in current mirrors, and biasing transistors. a- bias current shift. b- a change in the slope of the I-V curve due to Early voltage mismatches.

8.5 Summary

This chapter presented the important aspects in the design of the second insect vision chip. The testability and reconfigurability features, which facilitate the test of analog circuits, were described in Section 8.1.

Several novel circuits, which were implemented in various parts of the chip, were presented. The multisensitivity photodetector, described in Section 8.2, illustrated an increased sensitivity in the response of the photodetector by two orders of magnitude. In Section 8.3, it was shown that the MNC circuit can reduce the 100 Hz AC noise by a factor of ten, and in Section 8.4, it was shown that by using the CLM-OTA circuit, which uses the channel length modulation effect in MOS transistors, very low transconductance values can be achieved. The CLM-OTA has been used in the TCD circuit, for detecting the temporal variations of the light intensity.

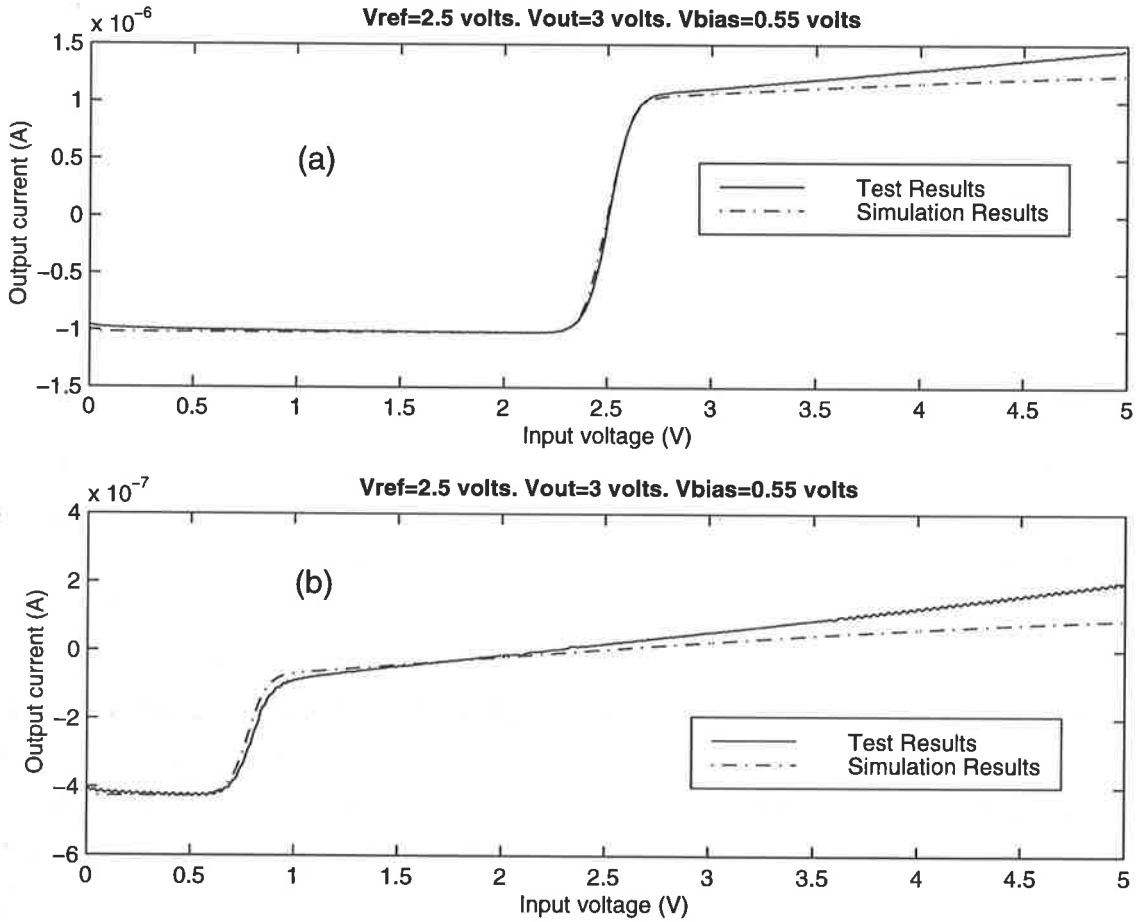


Figure 8.14: Test result of the OTA circuit. a) The measured and simulated output in the simple-OTA mode. b) The measured and simulated output in the CLM-OTA mode.

Chapter 9

Bugeye V: The Fifth Insect Vision Motion Detection Chip

Previous generations of the Bugeye motion detection chips represent designs which focused on circuit level implementations, mainly to comply with the analog VLSI (AVLSI) design framework. As these designs evolved, the circuits required for the implementation of the template model were individually perfected, and all exceeded the required characteristics. However, the operation of the circuits in an array of motion detection cells was severely affected by mismatch and digital noise (see Section 6 for a description of these issues).

The goal in designing Bugeye V was to provide a useful device for system level integration into projects, such as robot navigation and collision avoidance. For this reason, reliability, flexibility, and programmability were the primary design issues. Consequently, the use of analog VLSI circuits was limited only to a contrast enhancement stage using the MNC circuit, and digital circuits were used in the implementation of the processing and interface modules.

Bugeye V comprises all the elements necessary for the implementation of the template model. In addition, it has several other functions for facilitating its interfacing to other devices, such as general purpose microcontrollers and personal computers.

9.1 Architecture

The architecture of Bugeye V is depicted in Figure 9.1. Dashed arrows represent the control signals, and other arrows show the direction of data flow. There are two separate photocircuit arrays, one with 8×64 and the other with 4×64 pixels. The output of the detector array is digitized using an 8-bit ADC. All the processing stages in the template model, including temporal contrast detection, template formation, and template matching, are performed on these digitized signals. A 512×10-bit SRAM is used to store the image information (8 bits), and contrast information (2 bits) of each pixel in a a frame. The data from all intermediate stages

can also be read out separately.

For temporal contrast detection, the intensity value of the pixel in the previous frame, which is stored in the RAM, is read out and compared with the current value. After the templates are formed, they are compared with eight reliable templates, four of which represent motion to the right, and the other four motion to the left. These reliable templates, shown in Figure 9.2, have been found through analysis of the template model in the presence of noise [Nguyen 96]. All other templates are assigned a "no-motion" value. The *Global Motion Detection* module simply integrates and compares the number of templates indicating leftward and rightward motion.

The chip can be operated directly by providing it with the proper *Address*, *Data*, and *Control* signals, or through the *Parallel port* module, where most of the digital signals are generated inside the chip. This was necessary, as the parallel port of a PC, and most available microcontrollers, only support a limited number of digital I/O signals. The *Address*, *Data*, and *Control* buses are mainly used for testing individual modules.

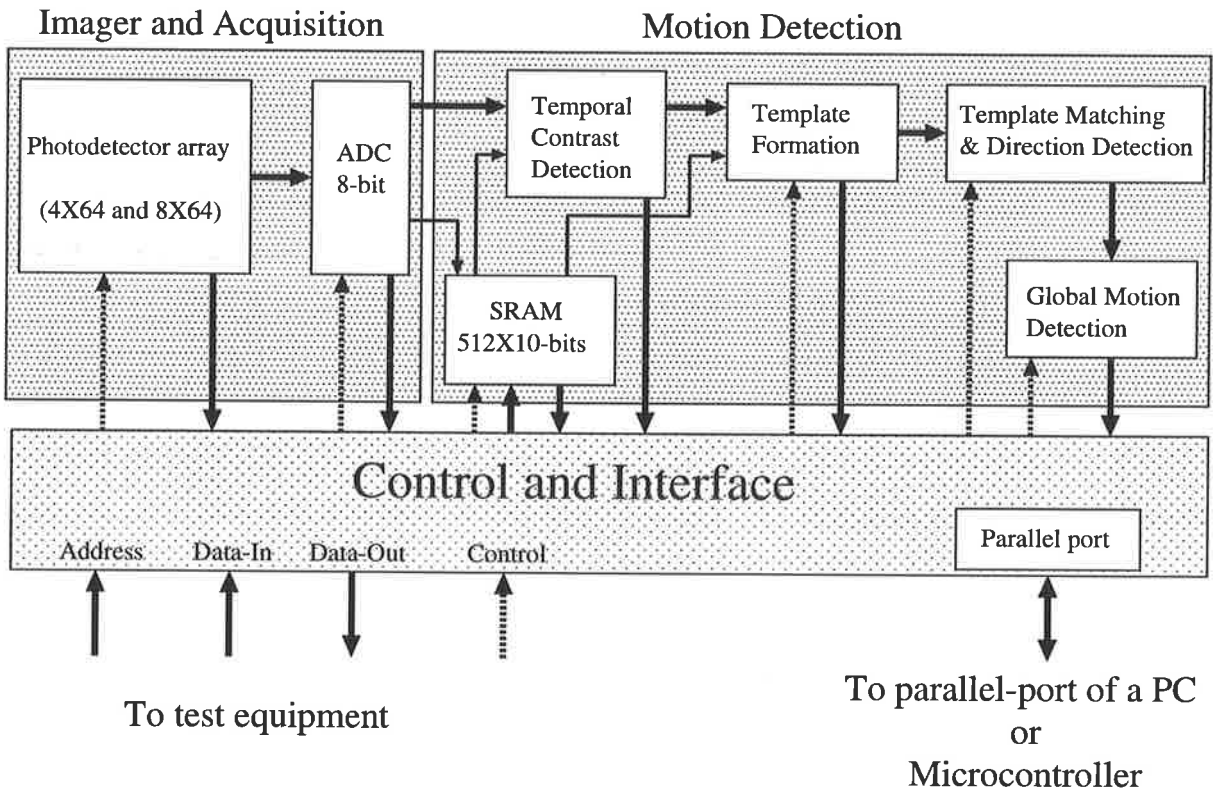


Figure 9.1: The architecture of Bugeye V.

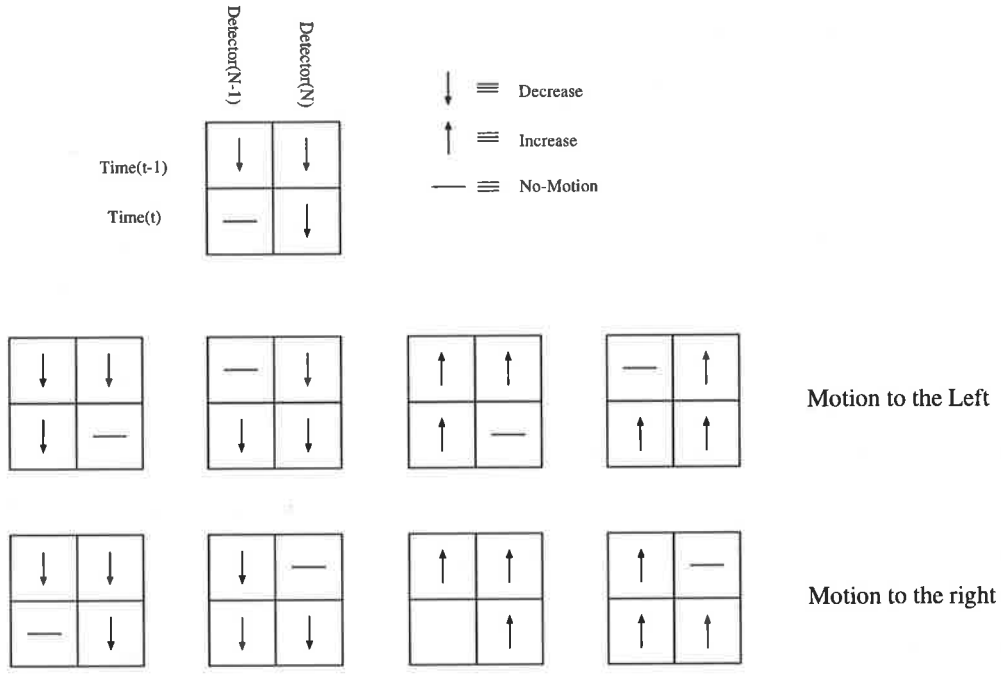


Figure 9.2: Eight reliable templates indicating motion to the left and to the right.

9.2 Photocircuits in Bugeye V

Two different photocircuits are implemented in Bugeye V. The first only comprises the photodetector, and the charge integration and sample-&-hold circuit shown in Figure 9.3 (see Section 5.2.6). The photocurrent I_{photo} is integrated and sampled. Transistor M3 acts as an amplifier, and transistor M4 selects the pixel. The output current is sent off-chip. In the simplest read-out circuit a pull-down resistor can be used, to transduce the current into voltage. Note that this photocircuit is in fact an active pixel sensor.

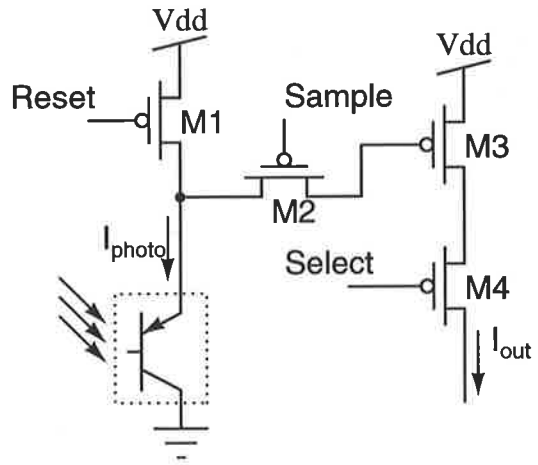


Figure 9.3: Schematic of the first photocircuit in Bugeye V.

The second photocircuit implements the "division by spatial average" model (see Section 2.3.1), which was also implemented in Bugeye II to reduce the multiplicative noise. In Bugeye V, this operation is used for contrast enhancement and dynamic range compression, as well as multiplicative noise cancellation.

There is a slight difference between the implementation of the spatial smoothing circuit in Bugeye II and Bugeye V. The circuit used in Bugeye II has a fixed smoothing window and uses more transistors than the circuit in Bugeye V (see Figures 8.6 and 5.20). The detailed schematic diagram of the second photocircuit used in Bugeye V is shown in Figure 9.4. The output current from the MNC circuit I_{MNC} is passed to an "integration and sample & hold" circuit, similar to that of the first photocircuit.

In the following tests the data has been acquired through the parallel port of a PC. The output of the array is sent to the on-chip 8-bit ADC, and digitized. The raw or processed image is then read out through the interface module.

Figure 9.5 illustrates the output image of the two arrays in response to a black strip in front of a white background. The mean luminance of the scene has been varied by about two orders of magnitude. The luminance of the black and white parts of the image are also different by about two orders of magnitude. Two interesting observations can be made:

- The simple photocircuit functions over a very small dynamic range. Although the integration time could be adjusted to accommodate variations of the mean luminance, it has been fixed to provide an insight into the merits of each circuit. The MNC-based photocircuit works over the entire dynamic range, without any need to changing the biasing voltages or integration period.

- While there are no visible signs of mismatch in the output of the simple photocircuit, the output of the MNC-based photocircuit exhibits a significant amount of mismatch. Reducing the mismatch by techniques similar to correlated double sampling [1] is not possible because the MNC process is nonlinear. Also as was seen in Section 6.1 mismatch is dependent on the spatial content of the image.

9.3 Motion Detection

Motion detection in Bugeye V is performed on the digitized image. The chip can output the raw image, the temporal variations, the templates, the accumulated number of leftward and rightward templates in a frame, and the global direction of motion. Since these operations are performed in the digital domain, performance is only limited by the imaging characteristics of

[1] In correlated double sampling (CDS) the offset value, when the pixel is reset, is subtracted from the pixel value. CDS is effective only when the offset, caused by mismatch, is linear.

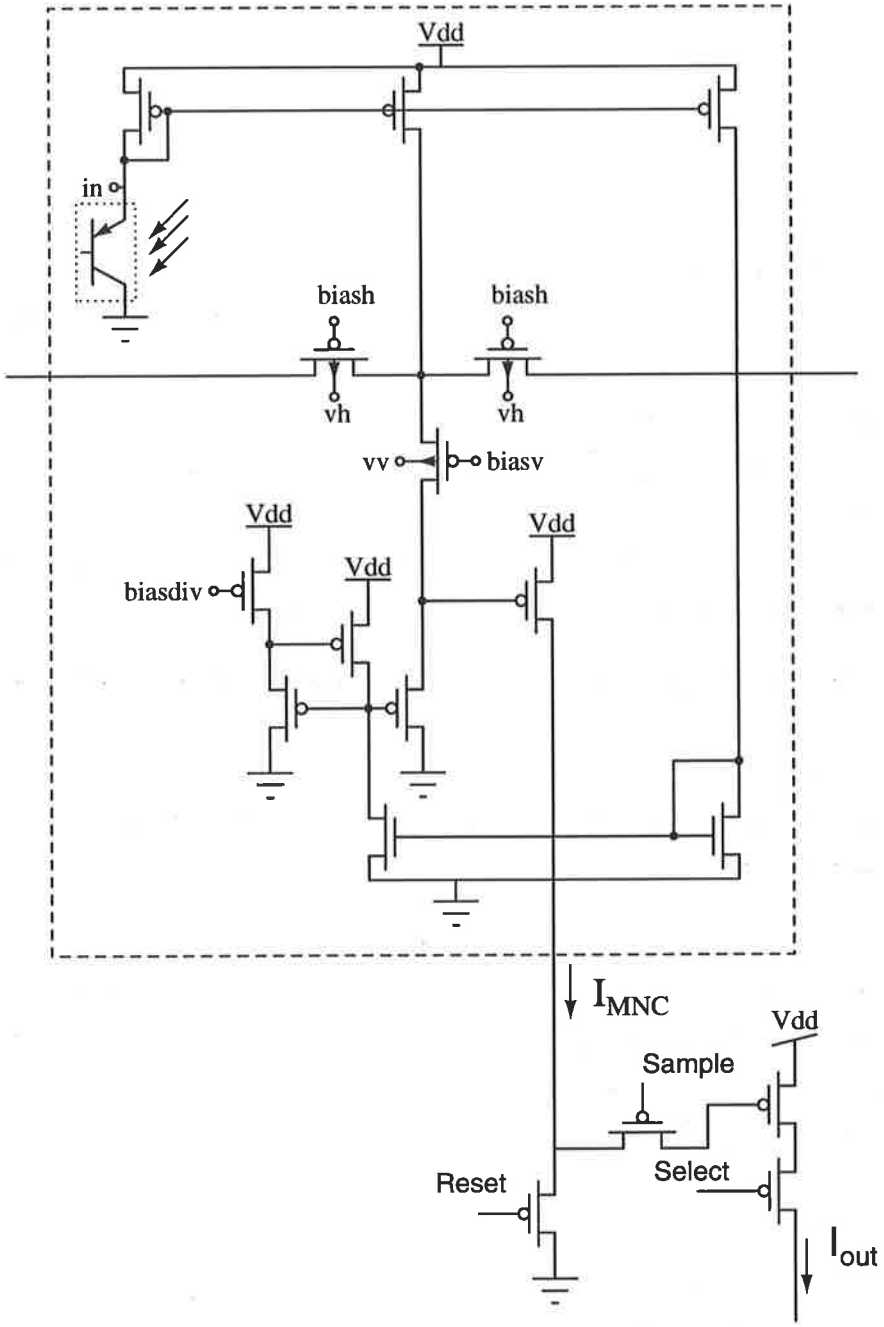
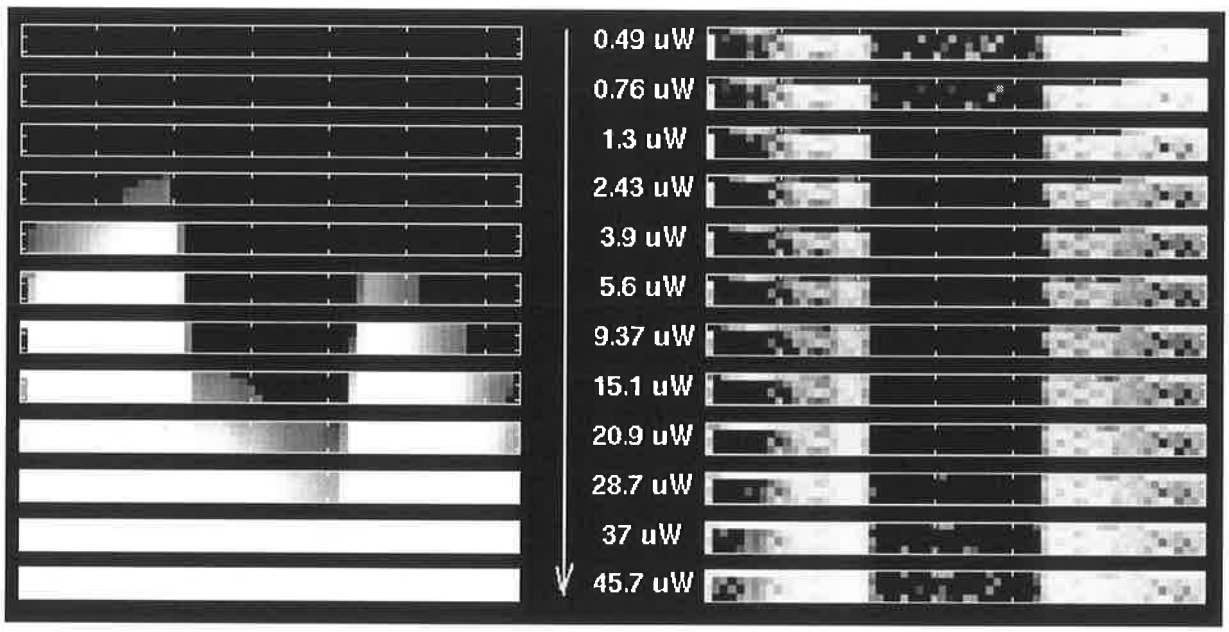


Figure 9.4: Schematic of the second photocircuit in Bugeye V.



(a) Light intensity (b)

Figure 9.5: Comparison between the output of the simple photodetector and MNC-based photocircuit. The input is a black strip on a white paper. a) The output image from the simple photocircuit array. b) The output image from the MNC-based photocircuit array. The numbers indicate the mean luminance.

the photocircuits, which are limited by noise and mismatch. Comprehensive study of the effect of noise has been done by X.T. Nguyen in his Ph.D. thesis [Nguyen 96], and here only typical outputs of the chip are presented.

Figure 9.6-a shows snapshots of a sequence of the raw image and of the template image from the simple photocircuit array. The stimulus is a hand waving in front of a white background. Both images have been acquired from the chip through the parallel port. Figure 9.6-b shows the image and output templates from the MNC-based photocircuits. Although the image of the hand is not as recognizable as the image from the simple photocircuit, the output templates are very similar. In the template images, the gray background represents "no motion", the black pixels correspond to templates indicating motion to the right, and white pixels correspond to templates indicating motion to the left.

9.4 Summary

This chapter presented the architecture and circuit level design of the fifth insect vision-based motion detection chip, Bugeye V. The aim of the design of this chip has been to provide a working device for real life applications, rather than to experiment with novel circuits. It thus has a programmable architecture, includes all necessary modules for system level integration, and can yield several types of image information, including raw image, and template image.

Section 9.1 presented the general architecture of Bugeye V and its function. Section 9.2 described the simple photocircuit and the MNC photocircuit, and presented the test results.

Section 9.3 illustrated the functionality of the chip when operated as a motion detector.

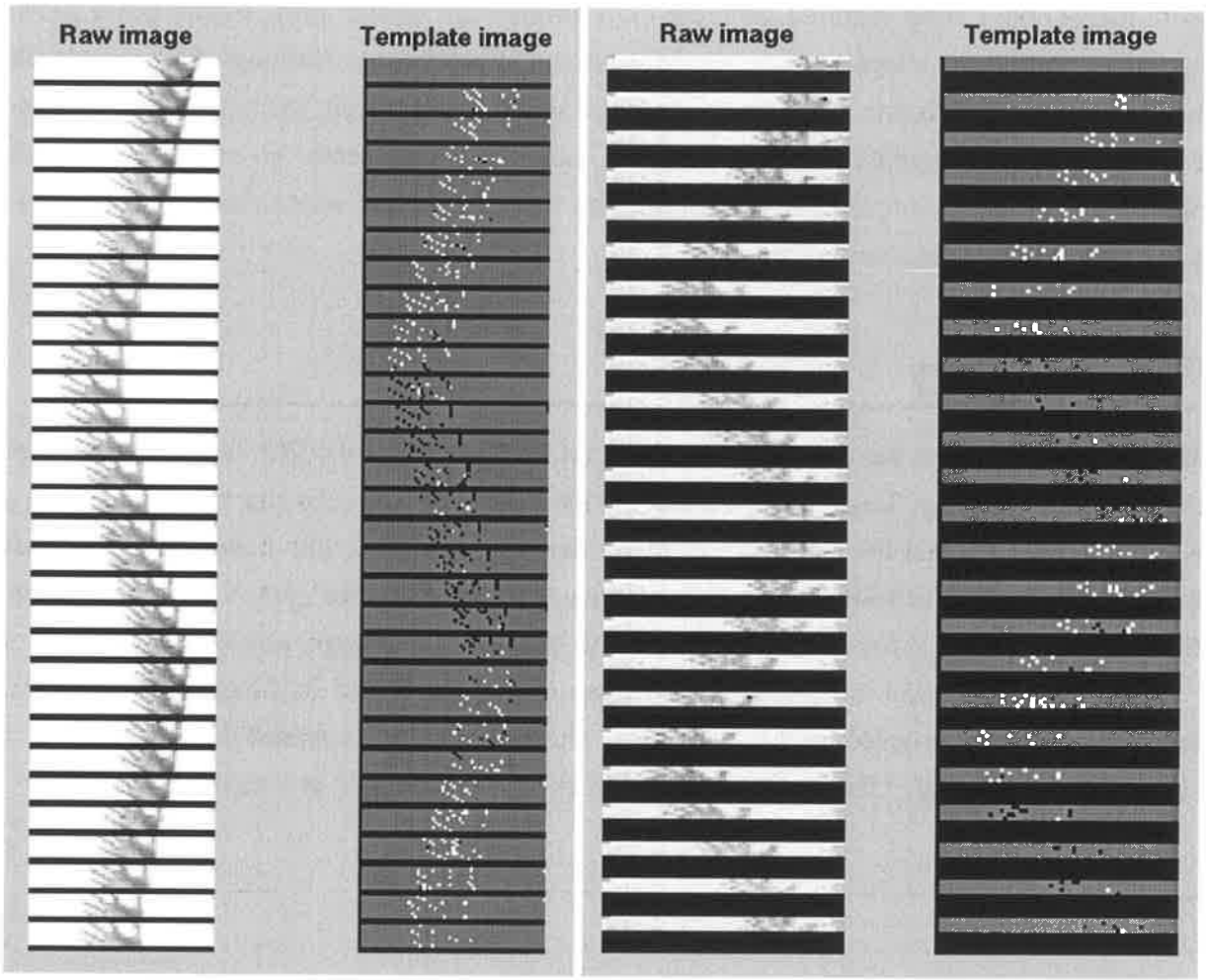


Figure 9.6: Motion detection in Bugeye V. a) Image and template output from the simple photocircuit. b) Image and template output from the MNC-based photocircuit. The gray background in template images represents no-motion, the black dots represent motion to the right, and the white dots represent motion to the left.

Chapter 10

MNCSI: Shunting Inhibition and Contrast Enhancement Vision Chip

Contrast enhancement is an important visual processing task in biological retina. Its task is to emphasize interesting features in the image, reduce noise, and make best use of available communication bandwidth by reducing the dynamic range of the image. Several computational and biological models for contrast enhancement were described in Section 2.3. VLSI implementations of contrast enhancement circuits were also presented in Section 5.3.5.

The MNCSI chip has been designed mainly as a test-bed for verifying the functionality of our implementation of two of the contrast enhancement models: shunting inhibition (SI), and multiplicative noise cancellation (MNC).

10.1 Shunting Inhibition

Shunting inhibition has been used to model both static (contrast enhancement) and dynamic (motion detection) properties of biological retina. Its simplicity makes it very attractive for VLSI implementations.

In addition to producing the general characteristics of contrast enhancement, SI can also simulate another property observed in biological retina, which is that the shape of receptive field changes with mean luminance. At normal light levels the function of the retina can be described by contrast enhancement models. However, at low light levels its function is more similar to spatial averaging.

In its simplest form, considering only nearest neighbor interaction, SI is described by:

$$\begin{aligned} \frac{de_i}{dt} &= I_i - be_i - ke_i(e_{i-1} + e_{i+1}) && \text{For feedback SI} \\ \frac{de_i}{dt} &= I_i - be_i - ke_i(I_{i-1} + I_{i+1}) && \text{For feed-forward SI} \end{aligned} \tag{10.1}$$

where e_i is the output of cell i, I_i is the input, b is a decay factor, and k is the inhibition factor. The feedback SI has been traditionally modeled by the circuit shown in Figure 10.1, as all the

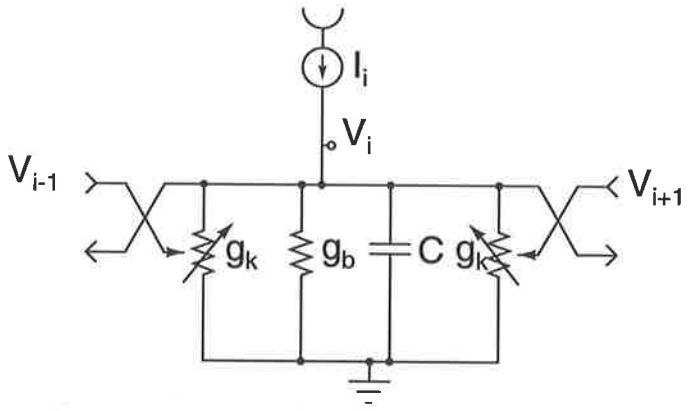


Figure 10.1: Voltage mode implementation of feedback SI.

parameters in the Equation 10.1 can easily be associated with component values in Figure 10.1. The only drawback of this representation arises from design issues related to the variable conductance g_k. In [Nilson et al. 94] a Gilbert multiplier has been used to implement this factor. However, Gilbert multipliers have a low dynamic range, and extending the circuit for interaction with more neighboring cells, e.g. in the case of 2-D arrays with rectangular or hexagonal tessellation, would require at least three extra transistors for each additional connection. In the current-mode approach, described later, only one extra transistor is needed.

10.1.1 Current Mode Implementation of SI

Current mode realization of various analog circuits is advantageous for several reasons, including increased dynamic range, broader design techniques (such as translinear and switched-current circuits), and simpler implementation of addition and subtraction operations [Tomazou et al. 90]. By rewriting Equation 10.1 for a current-mode design we can identify the building blocks needed for the implementation.

$$\frac{dJ_i}{dt} = I_i - \frac{I_b}{I_k} J_i - \frac{J_i}{I_k}(J_{i-1} + J_{i+1}) \tag{10.2}$$

where J represents the output currents, and I is the input current. The dJ/dt term in the left-hand side and the multiplicative term on the right-hand side require special circuits (see Figure 10.2). There are virtually no passive elements in current VLSI technologies which can be used to implement an inductor as required for the temporal differentiation of a current, except spiral inductors, which occupy a large area and can yield only very small inductances. However, current-mode temporal differentiation can be approximated by the circuit shown in Figure 10.3. In this circuit the input current is first delayed through the M1-M2 current mirror, which comprises a G_m-C delay element shown in the dashed box. The delayed current is then subtracted

from a copy of the input current using M3, M4, and M5 MOSFETs. The multiplicative term is implemented using the translinear multiplier/divider of Figure 8.7.

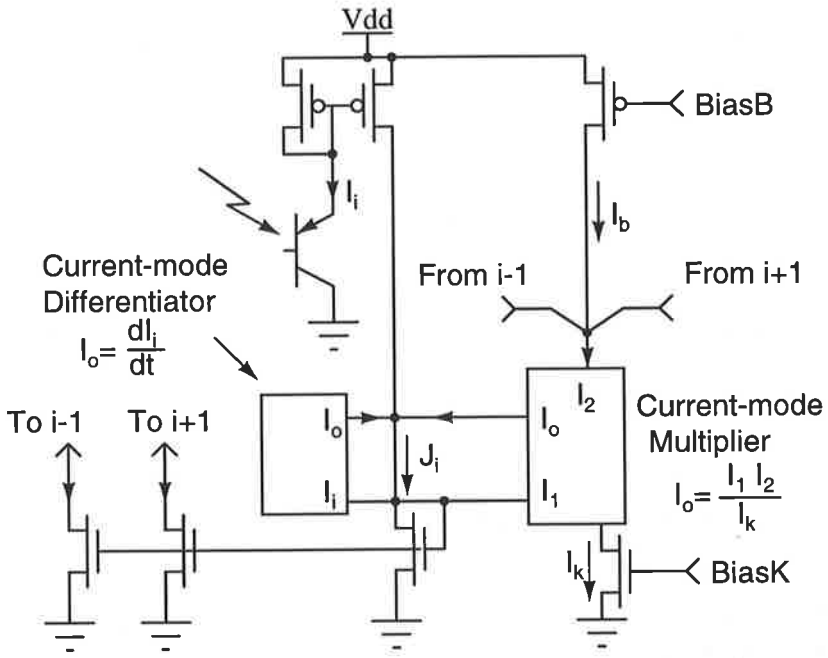


Figure 10.2: Schematic of the current-mode implementation of feedback SI.

10.1.2 Simulation and Test Results of the SI Circuit

The MNCSI chip is implemented in a standard 2 μ double-poly double-metal CMOS process. Several different circuits for the SI (with different photocircuits, and with and without the temporal differentiation term) were designed and laid out in a test chip. For test purposes only one-dimensional arrays with 64 cells were implemented.

Simulations were performed using HSPICE. Figure 10.4 presents a typical simulation output for a light bar with constant spatial contrast. The mean luminance has been varied over four decades. At low light intensities the circuit performs a spatial low-pass filtering and gradually changes its characteristics to a spatial band-pass filter, such that at higher intensities it performs an edge and contrast enhancement operation. This circuit clearly demonstrates the mean intensity dependent spatio-temporal behavior as observed in biological retinas [Pinter 84]. Figure 10.5 shows the same output, only at three mean luminance levels.

Tests were performed by focusing a point source on one element of the array. The rest of the array was exposed to uniform lighting. This background illumination was varied and the spatial response of the shunting inhibition array was measured. A typical output at three different background intensities are presented in Figure 10.6.

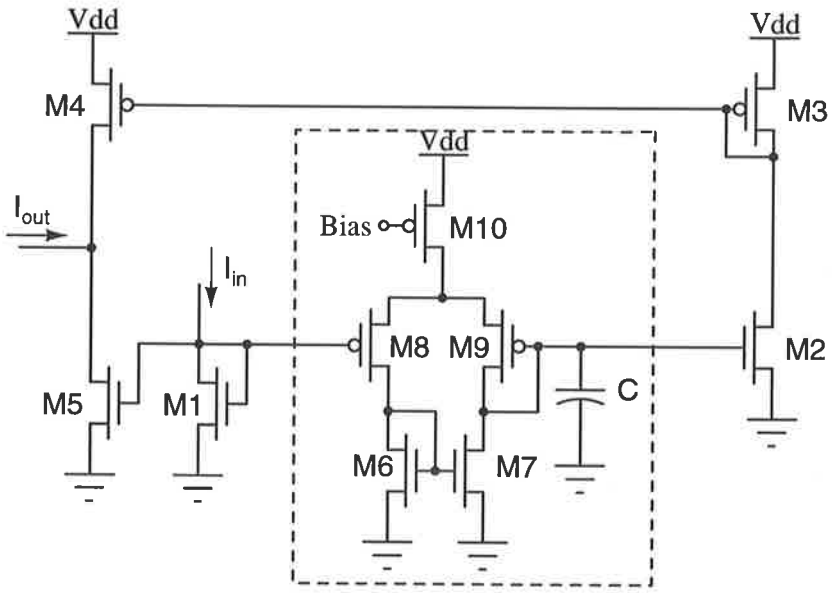


Figure 10.3: A current-mode time differentiator.

10.2 MNC Circuit

The performance of the MNC circuit as a contrast enhancement stage has been illustrated by its implementation in Bugeye V, which was presented in Section 9.2 [1]. Therefore, instead of restating the contrast enhancement properties of the MNC circuit, two other novel features implemented in MNCSI are described. The first feature is the use of the substrate node of the MOS transistors to increase controllability over the circuit operation. The second is the use of light-modulated MOS transistors to emulate the "mean luminance dependent receptive field size" property of biological retina.

10.2.1 Using the Substrate Node

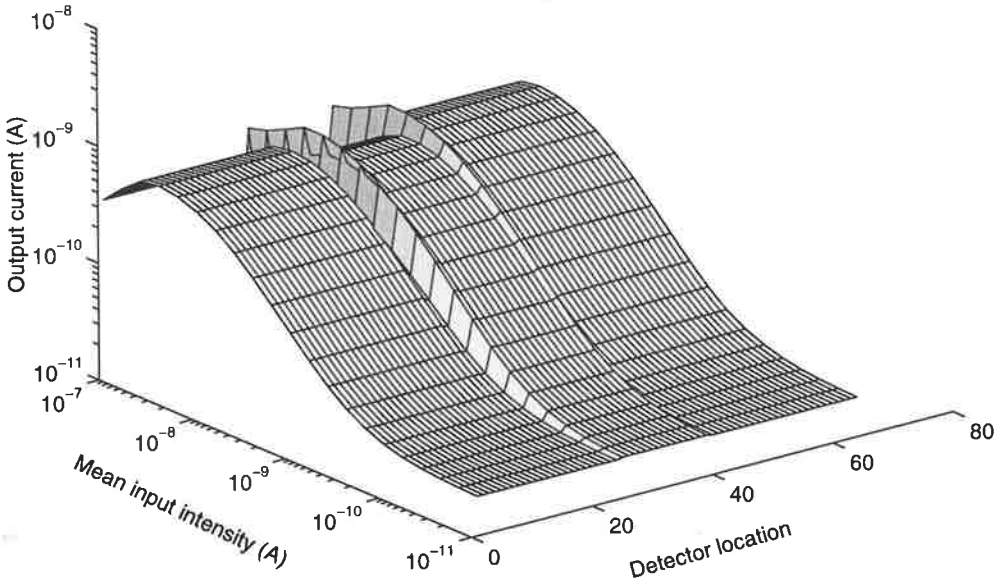
In a MOS transistor operating in the subthreshold region, the drain-source current is given by the following equation (repeated from Table 3.1).

$$I_{DS} = I_0 \frac{W}{L} e^{\frac{V_{GB}}{nU_T}} (e^{-\frac{V_{SB}}{U_T}} - e^{-\frac{V_{DB}}{U_T}}) \tag{10.3}$$

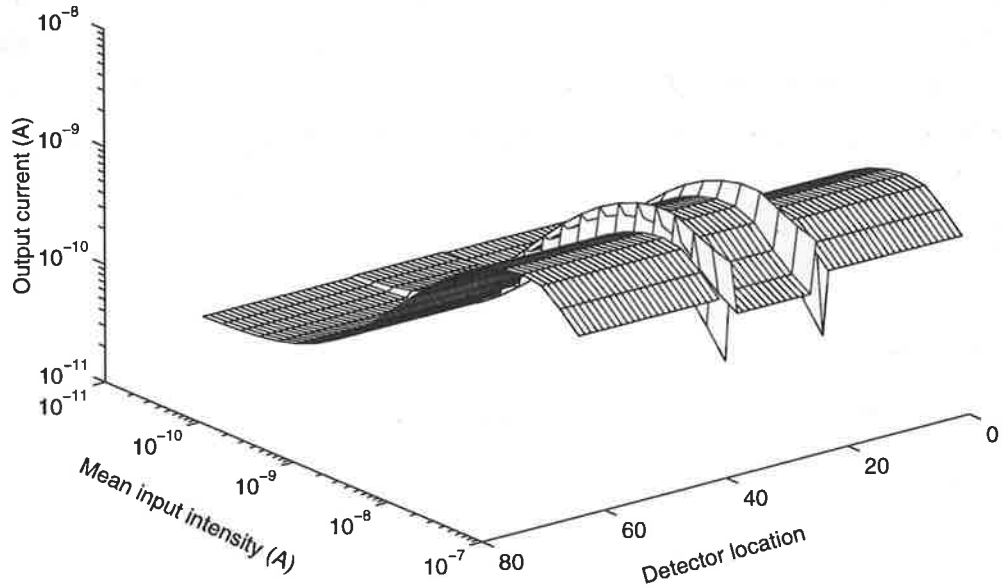
If $V_{DB} \gg U_T$, this equation can be simplified as

$$\begin{aligned} I_{DS} &= I_0 \frac{W}{L} e^{\frac{V_{GB}}{nU_T}} e^{-\frac{V_{SB}}{U_T}} \\ I_{DS} &= I_0 \frac{W}{L} e^{\frac{V_G}{nU_T}} e^{-\frac{V_S}{U_T}} e^{-\frac{(n-1)V_B}{nU_T}} \end{aligned} \tag{10.4}$$

[1] The design of MNCSI predates that of Bugeye V. However, Bugeye V was presented before MNCSI, so that its description follows Bugeye II.



(a)



(b)

Figure 10.4: Simulation result for the SI circuit. The output current at each detector location versus light intensity (given in the units of photocurrent, amperes). Both plots show the same output current at two different viewing angles.

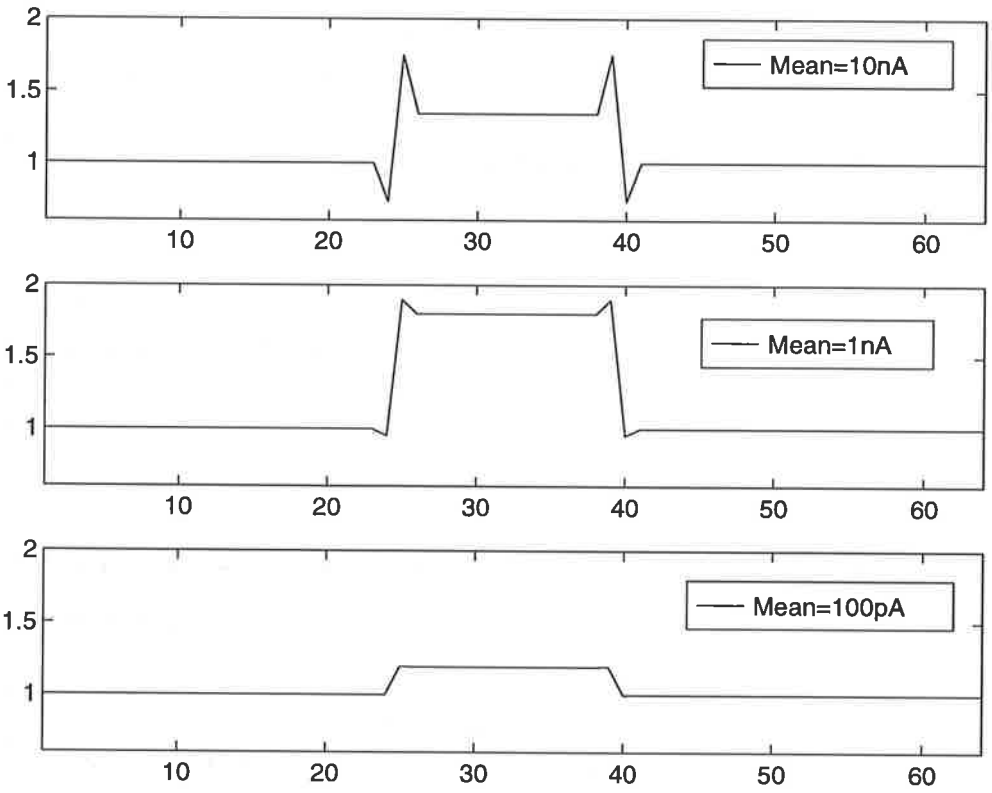


Figure 10.5: Simulation result for the SI circuit. Normalized spatial pattern of the output current at three different mean input intensities.

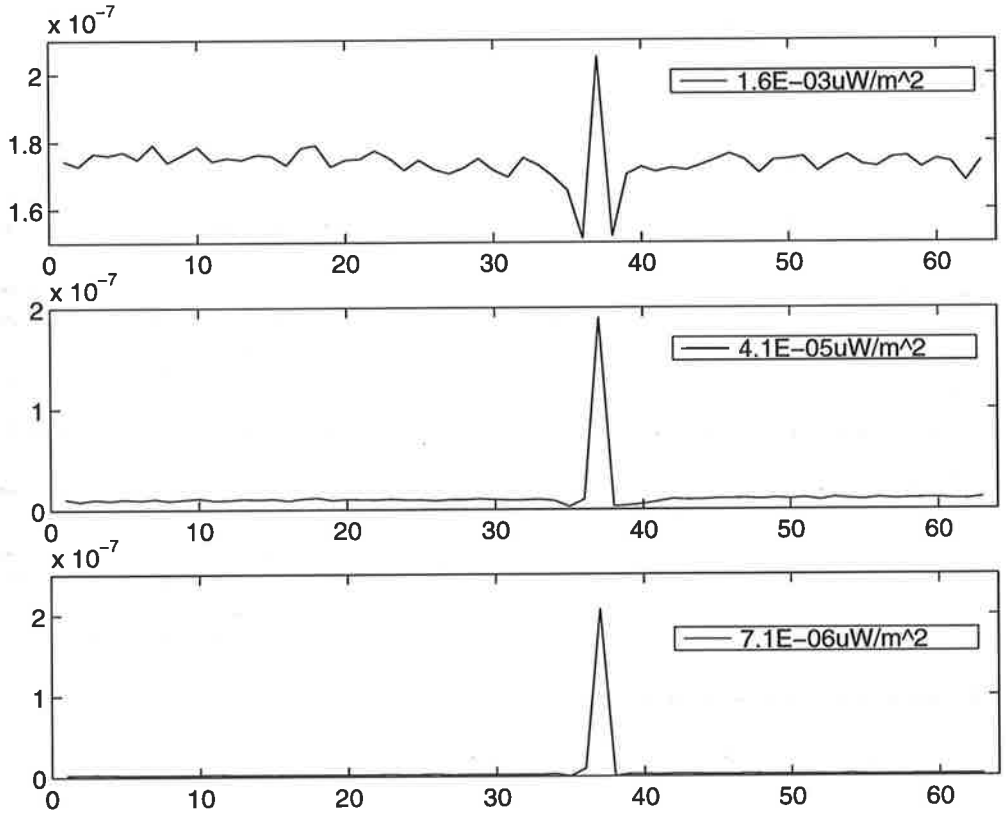


Figure 10.6: Test results for the SI circuit illustrating typical spatial response. As the intensity is increased the amount of inhibition is also increased.

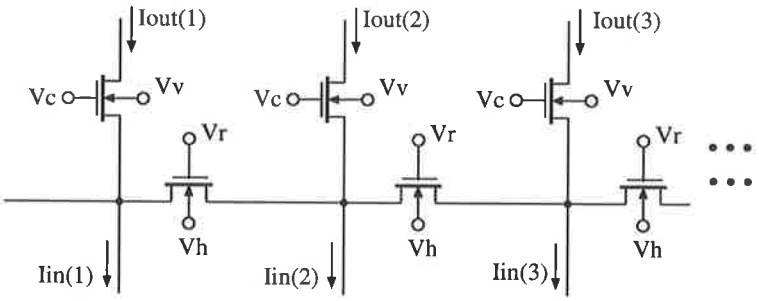


Figure 10.7: Diffusive smoothing circuit using substrate node voltages.

The parameter n is a process dependent factor, and is typically around 1.2 . For this value the transconductance of the transistor with respect to the substrate voltage is less than the transconductance of the gate or source, by a factor of $1.2/(1.2-1) = 6$.

This property can be used to increase the linear range of the circuits. [Sarpeshkar et al. 96b] describes such an OTA in which the input to the OTA is applied to the substrate node of the input transistors. In a simple 5-transistor OTA, the linear range of the I-V curve is about $4U_T \approx 100\ mV$. In the modified OTA, the linear range extends to more than 1 volt. It should be noticed that the substrate voltage should always satisfy the conditions $V_{DB} > -V_0$ and $V_{SB} > -V_0$, where V_0 is the turn-on voltage of the junction diodes.

In the MNC circuit, this property was used to increase the control over the adjustment of the width of the spatial smoothing window. Figure 10.7 shows the modified spatial smoothing circuit presented in Figure 5.20. The function of this circuit can be expressed by

$$K\Delta^2 I_{out} + I_{out} = I_{in} \tag{10.5}$$

where

$$K = e^{\frac{V_r - V_c}{nU_T}} e^{\frac{(n-1)(V_v - V_h)}{nU_T}} \tag{10.6}$$

Figure 10.8 shows the simulation results for this circuit, where the input is a spatial impulse. In Figure 10.8-a only V_r is varied and other voltages are fixed, and in Figure 10.8-b V_h is varied. The range over which the output is linearly controlled by the sweeping voltage is about 10 times larger in Figure 10.8-b compared to that observed in Figure 10.8-a. Figures 10.8-c and -d show profiles of the same plots.

The spatial smoothing circuit was used in the implementation of the MNC circuit, whose response to a spatial impulse, created by focusing a point source onto one of the photodetectors, is shown in Figure 10.9. It can be seen that the range over which the function of the MNC can be linearly controlled is about 1 volt.

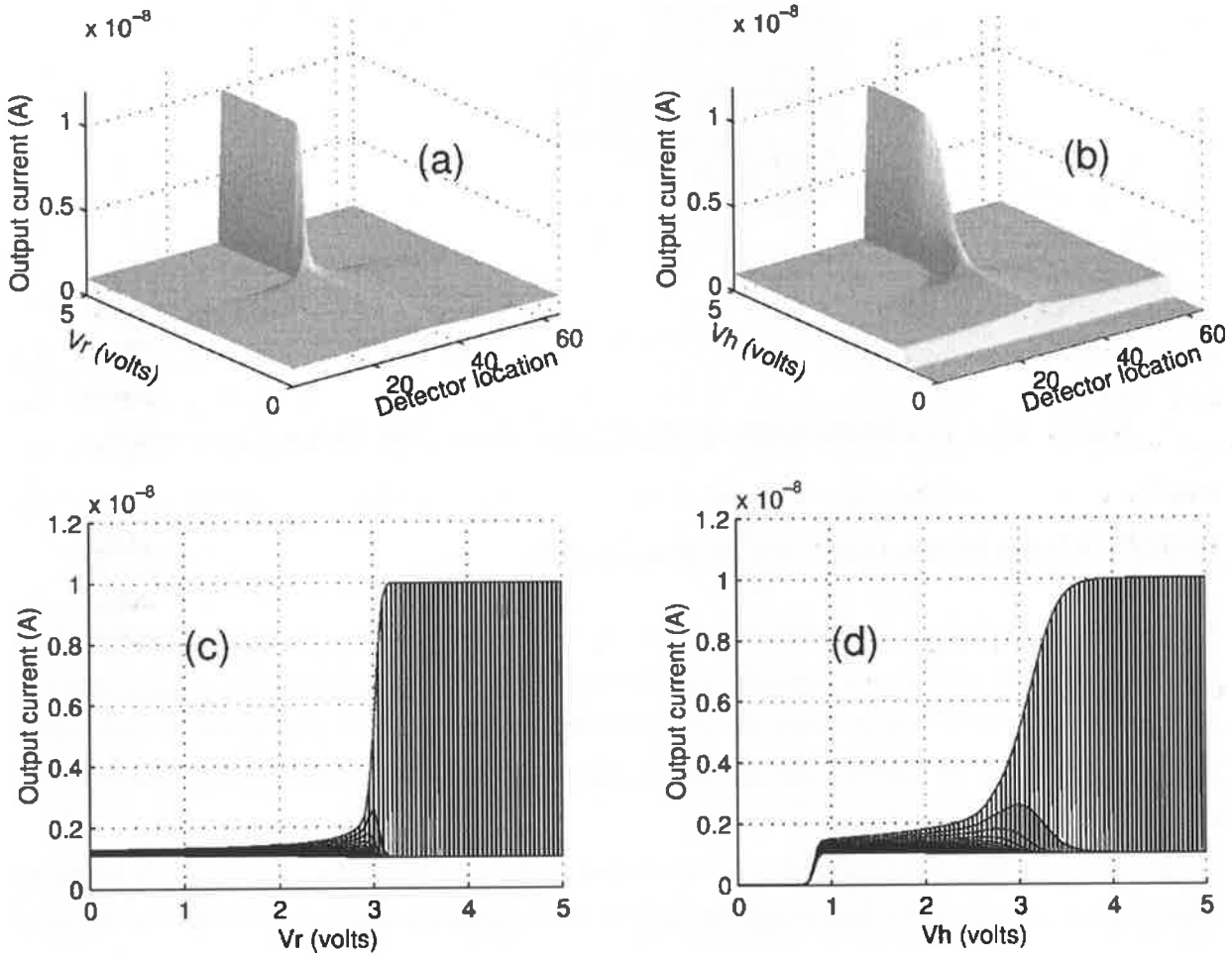


Figure 10.8: Spatial impulse response of the spatial smoothing circuit using substrate node voltage for varying the spatial span of the smoothing.

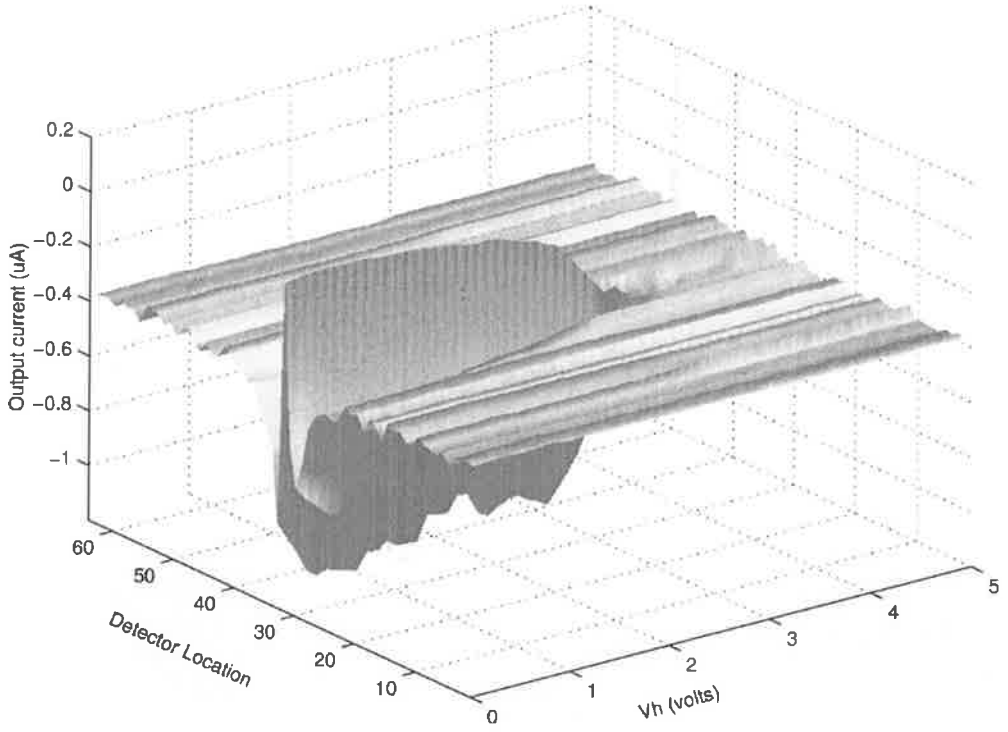


Figure 10.9: The spatial response of the MNC circuit. The voltage V_H is varied.

10.2.2 Light-Modulated MOS Transistors

One of the properties of the biological retina is the change of the size of the receptive field with mean light intensity. As the light level increases the size of the receptive field decreases. An implementation of this property by using shunting inhibition as a model was introduced in Section 10.1. Here, an intuitive model which utilizes the light dependent characteristics of the MOS transistors is presented.

There is a fundamental difference between this model and shunting inhibition. In shunting inhibition this property is obtained by the help of more complex functional blocks (in the form of multiplicative nonlinearity). It is also assumed that the circuit parameters are fixed. In the new approach, however, this property is obtained by using the inherent characteristics of the transistors in the circuit. In the spatial smoothing circuit shown in Figure 10.7, if the "resistivity" of the vertical transistors is modulated by light intensity, at lower light levels the vertical transistors exhibits more resistance than the transistors in the horizontal branches. Therefore, the spatial spread of the smoothing operation becomes wider. This circuit can be incorporated in the implementation of the MNC circuit as shown in Figure 10.10.

Figure 10.11 shows the output current of the MNC circuit, with and without the light-modulated transistor. The circuit diagrams of both circuits tested are exactly the same. However, in the layout of the circuit which uses the light-modulated transistors, the vertical transistor is exposed to light. This is achieved by opening a window in the Metal2 layer, which normally

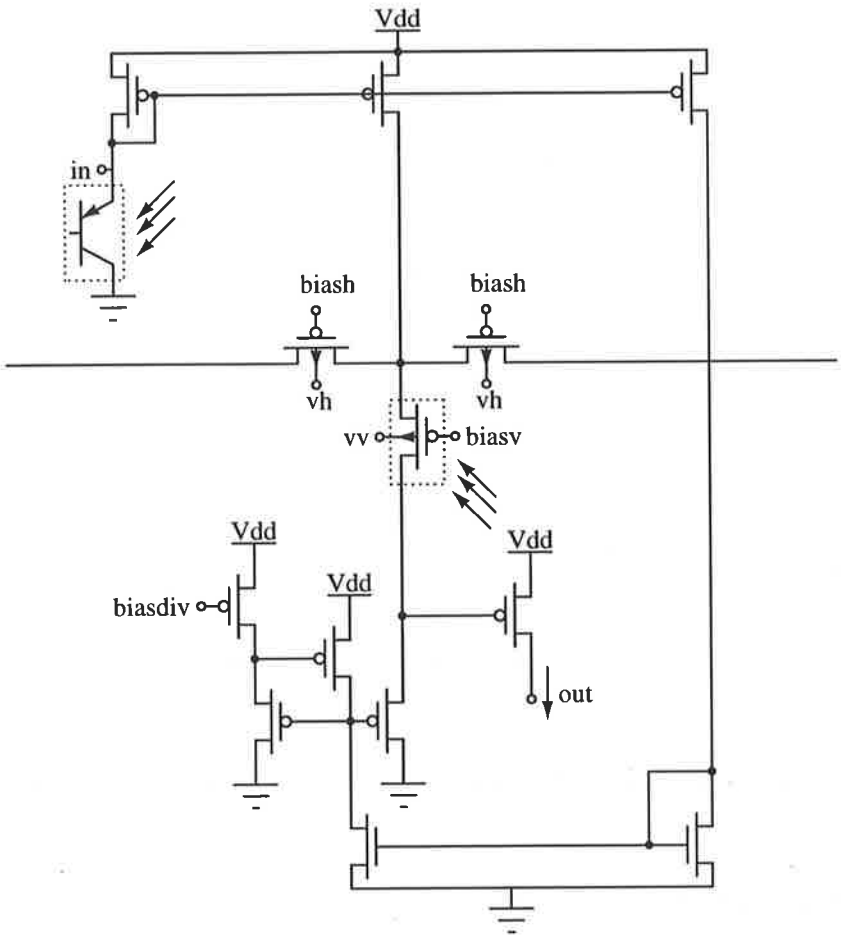


Figure 10.10: Circuit diagram of the MNC circuit utilizing light modulated transistors. The transistor in the dashed box is exposed to light. Other transistors are shielded by Metal2.

acts as a light shield on top of all the circuits.

10.3 Summary

This chapter described various circuits implemented in the MNCSI chip. In Section 10.1, shunting inhibition, its voltage mode and current mode implementation, and the simulation and test results of the implemented current mode circuit, were presented.

Two new ideas: the use of substrate node, and light-modulated MOS transistors, which have been used to improve the performance of the MNC circuit, were presented in Section 10.2. By using the substrate node as a control node, the linear range of the circuits can be increased, and by using light-modulated transistors, a straightforward implementation of the "light dependent receptive field" characteristic, observed in biological retinas, has been made possible.

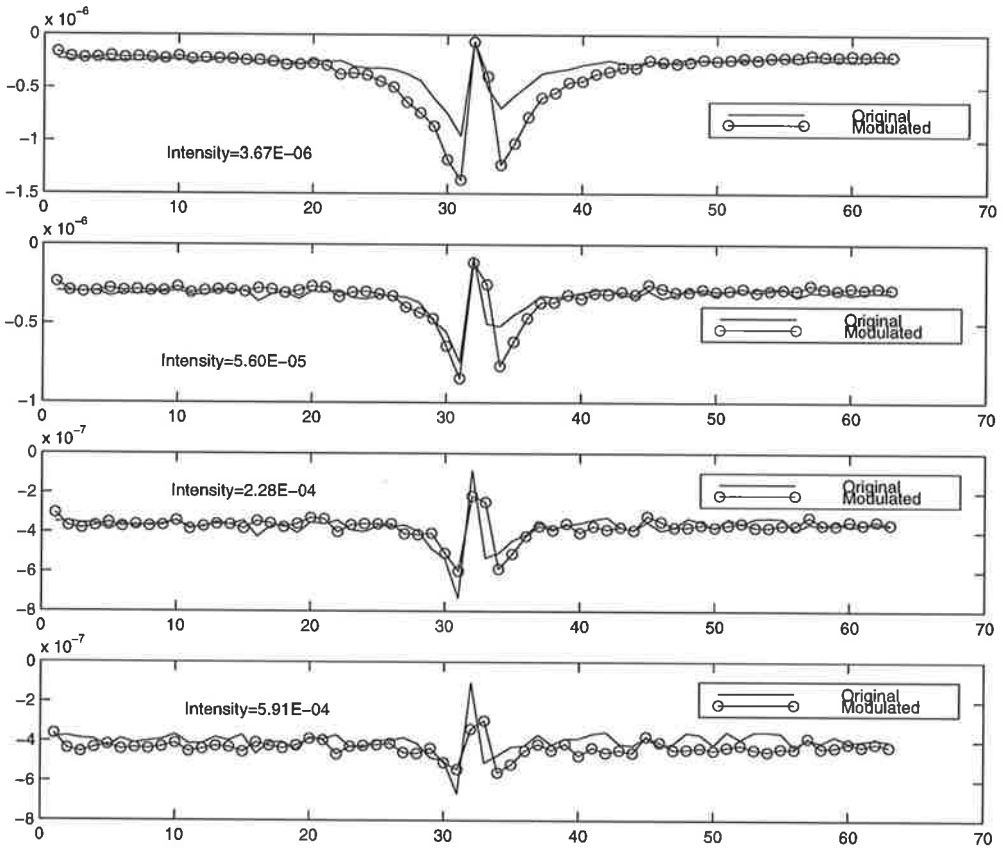


Figure 10.11: Output current of the MNC circuit with and without the light-modulated transistors.

Part IV

Discussions and Conclusion

Chapter 11

A Summary

In this dissertation many issues related to the analog VLSI (AVLSI) implementation of artificial and biological vision models were presented.

The first part addressed the mutual requirements of "AVLSI" and "computer vision". Biological and computational models of motion detection and contrast enhancement were presented. The offerings of the dominant VLSI technologies for implementing AVLSI vision chips were highlighted, and circuit design methodologies were presented.

The second part focused on the issues pertaining to the implementation of vision chips. Issues from pixel level to architectural levels were explained. Photodetectors, photocircuits, spatial processing circuits, and spatio-temporal processing circuits were presented. Two of the most dominant issues in AVLSI, i.e. mismatch and digital noise were addressed.

The third part described the specific implementation of three of the vision chips designed by the author, namely two motion detection chips, Bugeye II and Bugeye V, and a contrast enhancement vision chip, MNCSI. Bugeye II represents a bottom-up approach, where circuit level issues were given priority, and Bugeye V a top-down approach, where system levels issues were the primary concern.

Chapter 12

Contributions

This thesis provides insight into many issues related to AVLSI. It contributes to this area in several ways and at several levels. A systematic approach to AVLSI design is proposed, major problems are highlighted, and novel circuits and ideas for vision chips are presented. The salient points in this thesis are reiterated here.

- **A systematic approach to the design of vision chips.**
  A major concern of this thesis was the establishment of a systematic framework in which to present the knowledge in the area of vision chips. In Part II the basic material required for building such a framework was presented.

- **Highlighting the pitfalls of AVLSI.**
  Analog VLSI (using subthreshold circuits) has been regarded as the workhorse of vision chip design. Vision chips using this framework, however, have shown little success in providing reliable devices for vision processing, due mainly to the presence of mismatch and digital noise. Visual processing requires information across an image, whether local or global, and hence even if individual circuits function properly, the overall operation of the chip function of the network can be severely affected by mismatches between the circuits.

  Chapter 6 addressed the mismatch and digital noise problems. In particular, mismatch in analog networks for spatial processing was discussed, and it was shown that even for simple circuits, the amount of mismatch can be significant.

- **How is biological plausibility important?**
  Another important issue is the versatility and validity of the analogy that is made between VLSI technology and biological neural systems. The analogy is often made by stating two of the limitations of each technology, the accuracy and interconnection limitations. This analogy could be misleading in several respects.

Firstly, there exists no definitive evidence concerning the accuracy limitations of individual biological neurons, and whether their noise and accuracy levels are within the reach of electronic circuits.

Secondly, VLSI technology is far inferior to its biological counterpart. Biological retina has a 3D structure, and can change its morphology and topology (to some extent) for adaptation. By contrast, currently available VLSI technologies, have a 2D structure, and are fixed in terms of device position and size.

Thirdly, Biological retinas use special *devices* for each function. In VLSI technology there are only limited types of devices, and various functions should be constructed using these basic devices. The only advantage of VLSI devices is that they are about five to six orders of magnitude faster than biological neurons.

- **Tessellation: is it really important?**
  In many *silicon retinas* hexagonal tessellation has been used primarily to represent a biologically plausible structure. Section 4.1.2 argued this point, and it was concluded that unless the slight *theoretical* advantages of the hexagonal tessellation are essential to the performance of the vision chip, its use is not easily justifiable due to practical limitations in VLSI implementations. Moreover, both rectangular and hexagonal tessellations, with regular or irregular placements are seen in biological retinas. Therefore, using hexagonal structures only for the sake of making a biologically plausible sensor is not justified.

- **Circuits and networks for spatial processing.**
  Section 5.3 illustrated the manner in which simple and intuitive ideas can result in powerful and efficient circuits. It was also shown that a systematic path can also lead to such a circuit. Another conclusion made is that design ingenuity can result in even more area efficient circuits, even though there is no systematic way to achieve this.

- **Analysis of resistive networks.**
  A method for analyzing second-order resistive networks is presented in Appendix B. This method not only can yield the stability criteria for a resistive network with negative resistors, but also can easily provide the shape of the spatial convolution kernel of the network.

- **New circuits and ideas.**

  - **Multiplicative noise cancellation. (MNC)**
    In Section 8.3 a method for reducing the effect of 100 Hz AC noise was presented. It was shown that the MNC circuit could reduce the relative value of the AC noise by an order of magnitude, and at the same time exhibit contrast enhancement properties.
  - **A simple spatial smoothing circuit.**
    An intuitive spatial smoothing network was presented in Section 5.3 (see Figure 5.18).

- **Channel length modulation-based OTA (CLM-OTA).**
  By using the channel-length modulation property of MOS transistors, an OTA with very small conductance has been designed (see Section 8.4).
- **Using light modulated transistors.**
  In Section 10.2.2, the physical properties of a MOS transistor when exposed to light were utilized to implement the "luminance dependent receptive field size" property observed in biological retina.
- **Mixed photodetector structures.**
  In Appendix A, analysis of different photodetector structures revealed that a better response in the blue part of the spectrum (short wavelengths) can be obtained by mixing the lateral and vertical structures. Both bipolar and diode structures can benefit from this characteristic.
- **An implementation of shunting inhibition.**
  A current mode implementation of shunting inhibition was presented in Section 10.1. Both the static and dynamic components of the shunting inhibition equation were implemented.
- **Using the substrate node voltage.**
  In Section 10.2.1 it was shown that the linear range of operation of the spatial processing circuits can be increased considerably by utilizing the substrate node potential of a MOS transistor.

Chapter 13

Future Work

Research in the area of vision chip design can follow several paths. At the circuit design level, new circuits for performing various image processing operations in more efficient ways are required. At the architectural level, methods for improving system integration can be researched. And at the algorithm level, VLSI friendly algorithms for various vision processing tasks can be developed. Here, some of the major issues that were identified by this research as requiring further study and experimentation are highlighted.

- **Mismatch in VLSI circuits.**
  Mismatch is the most noticeable problem in analog VLSI design. Much work has been carried out on identifying the sources of mismatch and characterizing it. However, there is little literature concerning its effect on AVLSI networks.

  In order to reduce mismatch, conventional solutions suggest using larger devices, and higher operating currents. However, this becomes impractical in vision chips due to area limitations. Adaptive mechanisms are therefore necessary to change the parameters of individual devices. In an interconnected network of circuits two issues should be addressed.

  Firstly, algorithms which present adaptive properties should be investigated. These algorithms should be able to provide feedback to compensate for errors caused by device mismatch. Secondly, devices whose parameters can be tuned after process fabrication, should exist. While floating gate transistors are a good candidate for this purpose, their use is limited by, for instance, the need for high programming voltages. At the same time the performance of AVLSI networks in the presence of mismatch should be studied, in order to identify circuits which are less sensitive to mismatch.

- **Digital noise.**
  For many vision chip applications, such as motion detection, highly sensitive circuits are required to detect small variations in spatial or temporal contrast. Even though the design of such circuits is, in itself, relatively straightforward, problems arise when the

chip contains both analog and digital circuits, as the noise generated by digital switching can easily dominate weak input signal levels. Methods used for alleviating this problem consist of isolating the analog and digital circuits (which is not practical for vision chips), using differential signals (which is costly in terms of area), or decreasing the sensitivity of the circuits (which is counterproductive).

Other methods should be investigated with a view to restricting the amount of digital activity on the chip, especially in close proximity to sensitive circuits, and to develop circuits that are less sensitive to digital noise (by using decoupling mechanisms, for example).

- **New circuits.**
  A large proportion of research in the area of vision chip design, is still concerned with circuit design, and this research topic is still in its infancy. Part of the limitations are related to technological issues, such as area and interconnect density limitations. However, circuit design issues have been the most important parameter in the development of successful vision chips. Each different vision task often requires completely different circuits, which require extensive design, fabrication, and experimentation time. As this area progresses the library of *useful* circuits becomes more complete.

- **Extending the template model.**
  In our implementation of the template model, only motion direction is detected at each pixel. Velocity information may also be extracted from the templates. This has been addressed by A. Yakovleff and X.T. Nguyen in their Ph.D. theses [Yakovleff 96,Nguyen 96], where template tracking and velocity estimation architectures have been proposed. Moreover, image segmentation using the template information may be performed to determine the boundaries of objects.

- **Adding programmability.**
  Most circuits implemented in AVLSI systems have fixed functionalities. Practical aspects of the design often require some degree of programmability and flexibility, in order to perform different operations. For example, it may be required to perform motion detection on either a spatially smoothed image, or on a contrast enhanced image. In general, this requires that both functions be physically implemented and one be selected. This becomes impractical, when many functions need to be implemented, due to area limitation.

  Programmability, however, is only practical for general purpose architectures, and in the context of AVLSI, this only leaves several types of neural network architectures. For vision chips, cellular neural networks are the most flexible structure in terms of configurability (see Section 5.3.1 for details).

  These features, i.e. being programmable and general purpose, come with a severe area penalty. A 20×22 programmable CNN occupies about 30mm^2 in a 0.8μ m CMOS pro-

cess [Dominguez-Castro et al. 97], while an array of more than 150×150 contrast enhancement cells can be implemented in the same area [Andreou and Boahen 95].

Part V

Appendices

Appendix A

Quantum Efficiency of Photodetectors

Each of the photodetector structures illustrated in Figure 5.1 can be analyzed relatively easily. Here we provide an analysis for each device, and derive equations for the quantum efficiency as a function of the geometrical and metallurgical parameters of the devices. The simplifying assumptions made in all derivations in the following sections are:

- Abrupt junctions with rectangular depletion regions.
- One dimensional current flow. This would not be true for minimum size devices, where vertical and horizontal dimensions are comparable.
- No high-level injection. This becomes important for very high intensity applications, for example for furnaces or welding inspection.
- No degeneration in highly doped diffusion regions.
- No recombination in depletion regions.
- No surface recombination. This parameter is particularly important for lateral devices and for photogates, where there is a significant number of active carriers close to the surface. In vertical devices, the processes which determine the characteristics of the device depend only on the parameters of bulk semiconductor.
- No surface reflectance.
- No diffusion in the bulk substrate. This is important for near infra-red detectors, as most of the carrier generation happens close to the bulk substrate.

There are also some other assumptions made for each device which will be explained individually when treating each device.

In order to improve the consistency between the simulation results from the derived equations and real measured data, the above parameters should be taken into account. However, the

derived equations can still provide a good insight into the device operation, and illustrate the effect of different parameters on the quantum efficiency. Moreover, there are no accurate data available for the physical and metallurgical parameters in most processes, and hence these will be ignored here. In the extreme case one can use device simulation software to numerically derive the device characteristics.

A.1 Quantum Efficiency of a Vertical Junction Diode

For the structure shown in Figure A.1, the photocurrent is composed of two components: the drift current due to the drift of holes and electrons in the depletion region, and the diffusion current due to the diffusion of carriers outside the depletion region ([Moini 94]).

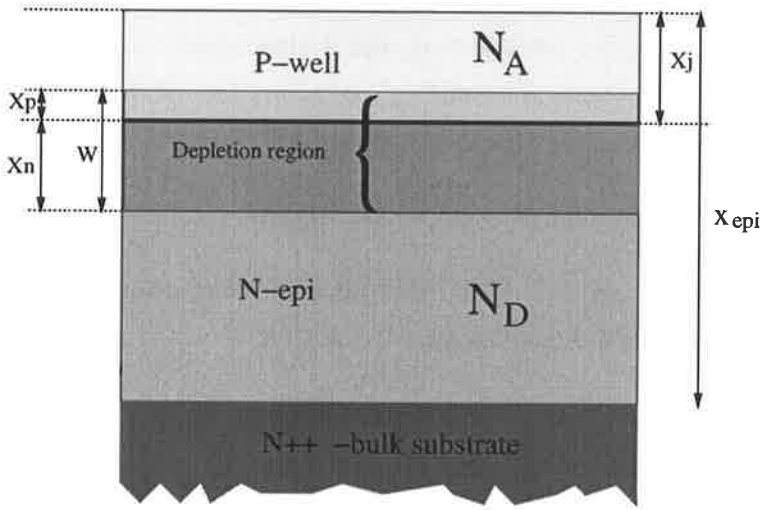


Figure A.1: The structure of a junction photodetector. x_j is the metallurgical junction depth, W is the width of the depletion region, and x_{epi} is the thickness of the epitaxial layer.

The drift current in the depletion region is:

$$J_{drift} = -q \int_{x_j - x_p}^{x_j + x_n} G(x) dx \tag{A.1}$$

where $G(x)$ is the carrier generation rate for an incident photon flux, Φ_0, in a semiconductor with an absorption coefficient of α, and is given by

$$G(x) = \Phi_0 \alpha e^{-\alpha x} \tag{A.2}$$

Hence

$$J_{drift} = q\Phi_0 e^{-\alpha(x_j - x_p)}(1 - e^{-\alpha W}) \tag{A.3}$$

x_n and x_p are the extent of the depletion region in the n and p sides of the junction and are given by

$$\begin{aligned} x_n &= \sqrt{\frac{2\epsilon(V_0+V_r)}{q}\left(\frac{N_D}{N_A(N_A+N_D)}\right)} \\ x_p &= \sqrt{\frac{2\epsilon(V_0+V_r)}{q}\left(\frac{N_A}{N_D(N_A+N_D)}\right)} \end{aligned} \tag{A.4}$$

where V_r is the reverse bias voltage applied to the junction, and V_0 is the built-in potential of the junction, and is equal to

$$V_0 = \frac{kT}{q}\ln\frac{N_A N_D}{n_i^2} \tag{A.5}$$

The diffusion component of the current can be found from the diffusion equation:

$$\begin{aligned} D_p\frac{\partial^2 p_n}{\partial x^2} - \frac{p_n - p_{n0}}{\tau_p} + G(x) = 0 \quad &\text{in the N-substrate} \\ D_n\frac{\partial^2 n_p}{\partial x^2} - \frac{n_p - n_{p0}}{\tau_n} + G(x) = 0 \quad &\text{in the p-well} \end{aligned} \tag{A.6}$$

where D_n and D_p are the diffusion coefficients of the minority carriers, τ_p and τ_n are the lifetime of excess carriers, and p_{n0} and n_{p0} are the equilibrium minority carrier densities. The above equation can be solved under the boundary conditions $p_n|_{x=x_{epi}} = 0$, $p_n|_{x=x_j+x_n} = 0$, $n_p|_{x=0} = n_{p0}$, and $n_p|_{x=x_j-x_p} = 0$ to obtain

$$\begin{aligned} p_n(x) &= p_{n0} + Ae^{\frac{x}{L_p}} + Be^{-\frac{x}{L_p}} + Ce^{-\alpha x} \\ n_p(x) &= n_{p0} + De^{\frac{x}{L_n}} + Ee^{-\frac{x}{L_n}} + Fe^{-\alpha x} \end{aligned} \tag{A.7}$$

where L_p and L_n are the diffusion lengths of excess carriers, and

$$A = \frac{(-Ce^{-\alpha x_{epi}} - p_{n0})e^{-\frac{x_j+x_n}{L_p}} + e^{\frac{-x_{epi}}{L_p}}(p_{n0} + Ce^{-\alpha(x_j+x_n)})}{2\sinh\frac{x_{epi}-(x_j+x_n)}{L_p}}$$

$$B = \frac{(-Ce^{-\alpha x_{epi}} - p_{n0})e^{\frac{x_j+x_n}{L_p}} + e^{\frac{x_{epi}}{L_p}}(p_{n0} + Ce^{-\alpha(x_j+x_n)})}{-2\sinh\frac{x_{epi}-(x_j+x_n)}{L_p}}$$

$$C = \frac{\Phi_0 \alpha L_p^2}{D_p(1-\alpha^2 L_p^2)}$$

$$D = \frac{F\left(e^{-\alpha(x_j-x_p)} - e^{-\frac{(x_j-x_p)}{L_n}}\right) + n_{p0}}{-2\sinh\frac{(x_j-x_p)}{L_n}} \tag{A.8}$$

$$E = \frac{F\left(e^{-\alpha(x_j-x_p)} - e^{\frac{(x_j-x_p)}{L_n}}\right) + n_{p0}}{2\sinh\frac{(x_j-x_p)}{L_n}}$$

$$F = \frac{\Phi_0 \alpha L_n^2}{D_n(1-\alpha^2 L_n^2)}$$

The diffusion current can be expressed as:

$$J_{diff} = J_{diff,p} + J_{diff,n} = -qD_p\frac{\partial p_n}{\partial x}|_{x=x_j+x_n} + qD_n\frac{\partial n_p}{\partial x}|_{x=x_j-x_p}$$

$$J_{diff} = -q\frac{D_p}{L_p}Ae^{\frac{x_j+x_n}{L_p}} + q\frac{D_p}{L_p}Be^{-\frac{x_j+x_n}{L_p}} + qD_pC\alpha e^{-\alpha(x_j+x_n)} \tag{A.9}$$

$$+q\frac{D_n}{L_n}De^{\frac{x_j-x_p}{L_n}} - q\frac{D_n}{L_n}Ee^{-\frac{x_j-x_p}{L_n}} - qD_n\alpha Fe^{-\alpha(x_j-x_p)}$$

which can be simplified to:

$$J_{diff} = q\frac{D_p}{L_p}p_{n0}\frac{1-\cosh K_p}{\sinh K_p} + q\frac{D_p}{L_p}C\frac{e^{-\alpha x_{epi}}}{\sinh K_p} + qCD_pe^{-\alpha(x_j+x_n)}\left(\alpha - \frac{\cosh K_p}{L_p\sinh K_p}\right) +$$

$$q\frac{D_n}{L_n}n_{p0}\frac{1-\cosh K_n}{\sinh K_n} + q\frac{D_n}{L_n}F\frac{1}{\sinh K_n} - qFD_ne^{-\alpha(x_j-x_n)}\left(\alpha + \frac{\cosh K_n}{L_n\sinh K_n}\right) \tag{A.10}$$

$$K_p = \frac{x_{epi}-x_j-x_n}{L_p} \quad K_n = \frac{x_j-x_p}{L_n}$$

The parameters D_n, D_p, τ_n, and τ_p can be derived from the following empirical formulas for

silicon, as a function of impurity densities

$$\begin{aligned}
\tau_p &= \frac{1}{7.8 \times 10^{-13} N_D + 1.8 \times 10^{-31} N_D^2} \\
D_p &= \frac{kT}{q}\left(370 + \frac{370}{1+1.563\times10^{-18} N_D}\right) \\
\tau_n &= \frac{1}{3.45 \times 10^{-12} N_A + 9.5 \times 10^{-32} N_A^2} \\
D_n &= \frac{kT}{q}\left(232 + \frac{1180}{1+1.125\times10^{-17} N_A}\right)
\end{aligned} \tag{A.11}$$

The total current J_{opt} is the sum of the drift and diffusion currents.

$$J_{opt} = J_{drift} + J_{diff} \tag{A.12}$$

The above equations may be simplified for single-sided and shallow junctions to provide a better understanding of the effect of different parameters on the photoresponse of the device, but we keep them in their general form. The measured absorption coefficients for silicon are shown in Figure A.2. Typical parameters of a p-well–substrate and a diffusion–well silicon junctions are shown in Table A.1. The simulated quantum efficiency, J_{opt}/Φ_0, for these devices is plotted in Figure A.3. The quantum efficiency of the diffusion-substrate junction is higher than for the other two structures, and also spans a wider spectrum.

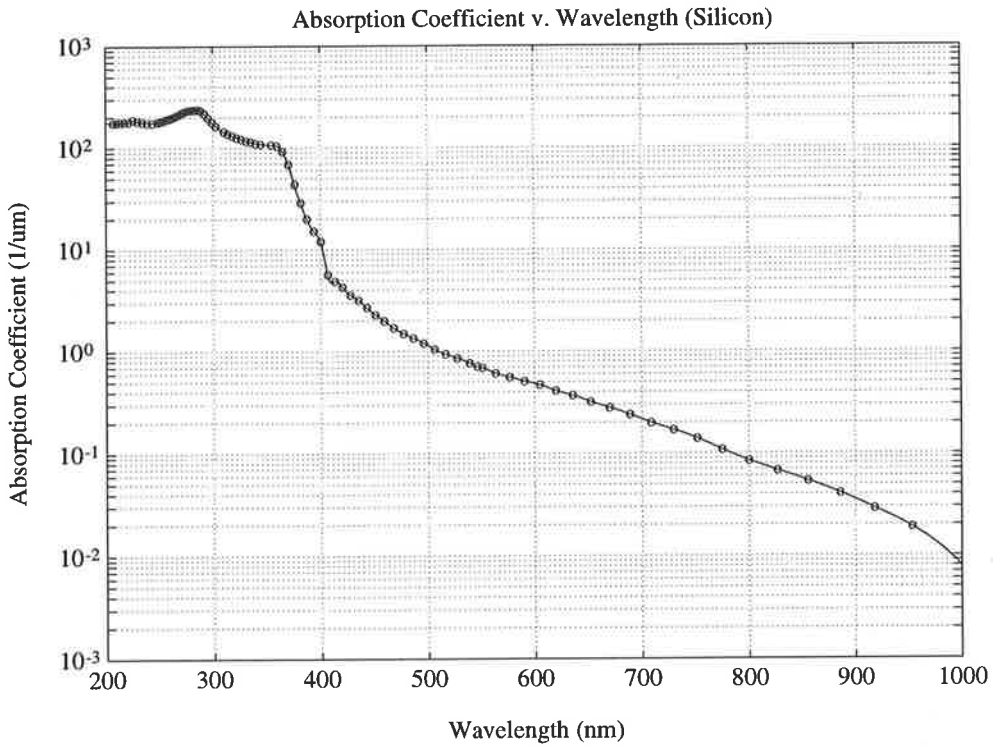


Figure A.2: Measured absorption coefficient of silicon.

Table A.1: Typical parameters of silicon junctions in a 2μm p-well standard process provided by Orbit Semiconductor Inc.

Diode structure	x_j μm	x_{epi} μm	N_D $1/cm^3$	N_A $1/cm^3$	V_{bias} volts	n_i $1/cm^3$	L_n μm	L_p μm
p-well–substrate	2.25	10–15	5.07×10^{15}	2.22×10^{16}	0	1.45×10^{10}	199.6	694
n-diff–p-well	0.47	2.25	2.22×10^{16}	1×10^{20}	0	1.45×10^{10}	199.6	0.71
p-diff–substrate	0.47	10–15	1×10^{20}	4.37×10^{15}	0	1.45×10^{10}	446.8	0.289

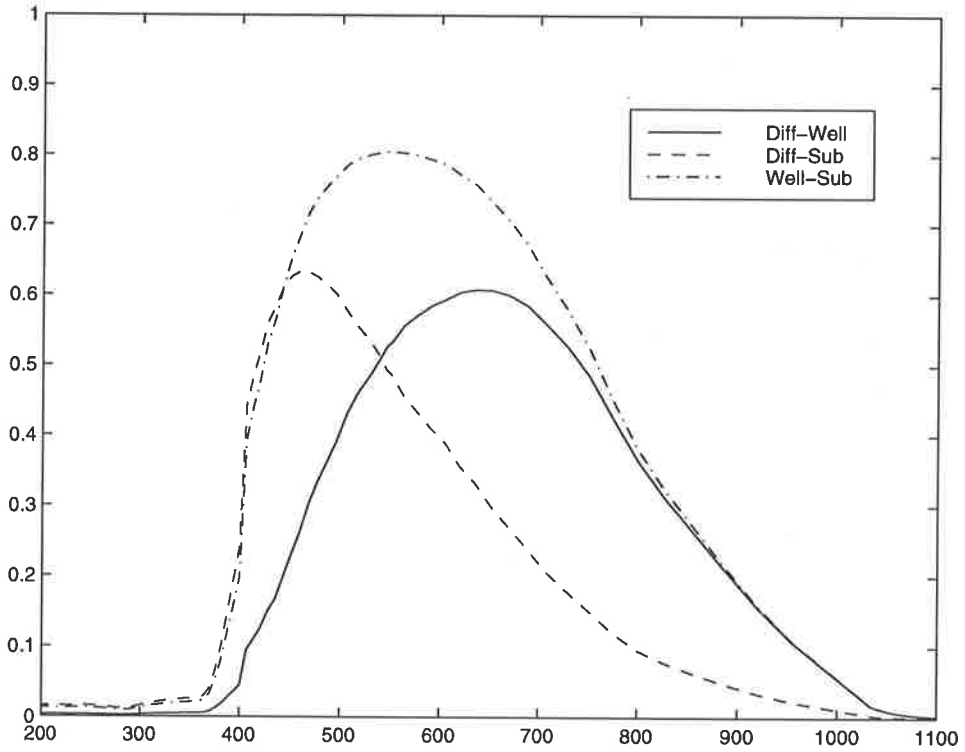


Figure A.3: Simulated quantum efficiency versus wavelength for three different junction diodes in a 2μm process.

A.2 Quantum Efficiency of a Lateral Junction Diode

The structure of a lateral photodiode is shown in Figure A.4. For analysis purposes, a few simplifying assumptions are made. Firstly, only the area between the two diffusion regions is assumed to be exposed to light. Otherwise, there will be a large contribution from the vertical bipolar component formed by p-diffusion/n-well/p-substrate, whereas in reality, the photogenerated electron-hole pairs will diffuse to other areas. Secondly, it is assumed that the effective depth of the device is only y_j, as again there will be some currents diffusing through other areas.

The diffusion equations in the P+ and N-well are

$$\begin{aligned} D_p\frac{\partial^2 p_n}{\partial x^2} - \frac{p_n - p_{n0}}{\tau_p} + G(y) = 0 \quad & \text{in the N-well} \\ D_n\frac{\partial^2 n_p}{\partial x^2} - \frac{n_p - n_{p0}}{\tau_n} = 0 \quad & \text{in the P+} \end{aligned} \tag{A.13}$$

By taking into account the boundary conditions $p_n|_{x\approx x_j} = 0$ and $p_n|_{x=x_n} = 0$, we will have

$$\begin{aligned} p_n(x) &= p_{n0} - \tau_p G(y) + \frac{C(e^{-\frac{x_j}{L_p}} - e^{-\frac{x_n}{L_p}})}{2\sinh\frac{x_j - x_n}{L_p}} e^{\frac{x}{L_p}} - \frac{C(e^{\frac{x_j}{L_p}} - e^{\frac{x_n}{L_p}})}{2\sinh\frac{x_j - x_n}{L_p}} e^{-\frac{x}{L_p}} \\ C &= p_{n0} - \tau_p G(y) \\ J_{diff}(y) &= -qD_p\frac{\partial p_n}{\partial x}|_{x=x_n} = -\frac{qD_p(p_{n0} - \tau_p G(y))}{L_p \sinh\left(\frac{x_j - x_n}{L_p}\right)}\left[1 - \cosh\left(\frac{x_j - x_n}{L_p}\right)\right] \end{aligned} \tag{A.14}$$

The drift current is simply

$$J_{drift} = -q\Phi_0 G(y)(x_n + x_p) \approx -q\Phi_0 G(y) x_n \tag{A.15}$$

The total current can be obtained by integrating the addition of the drift and diffusion components across the depth and width of the device.

$$J_{total} = \int_0^{y_j} [A - (B + qx_n)\Phi_0 G(y)]dy = Ay_j + (B + qx_n)\Phi_0\{e^{-\alpha y_j} - 1\} \tag{A.16}$$

where

$$\begin{aligned} A &= \frac{qD_p p_{n0}\left(1 - \cosh\left(\frac{x_j - x_n}{L_p}\right)\right)}{L_p \sinh\left(\frac{x_j - x_n}{L_p}\right)} \\ B &= \frac{qD_p \tau_p\left(1 - \cosh\left(\frac{x_j - x_n}{L_p}\right)\right)}{L_p \sinh\left(\frac{x_j - x_n}{L_p}\right)} \end{aligned} \tag{A.17}$$

$$\begin{aligned} x_n &= \sqrt{\frac{2\epsilon(V_0 + V_r)}{q}\left(\frac{N_D}{N_A(N_A + N_D)}\right)} \\ V_0 &= \frac{kT}{q}\ln\frac{N_A N_D}{n_i^2} \end{aligned} \tag{A.18}$$

Figure A.5 shows the simulation result of this structure for a typical 2μm process. As expected, there is a large blue response because all the carriers generated close to the surface are absorbed by the device. The poor response at longer wavelengths is due to the fact that we have considered the contribution of those carriers which are up to $y - j$ deep into the device, which is very shallow. This structure can be combined with the vertical photodiode, by exposing all sides of the diode to light.

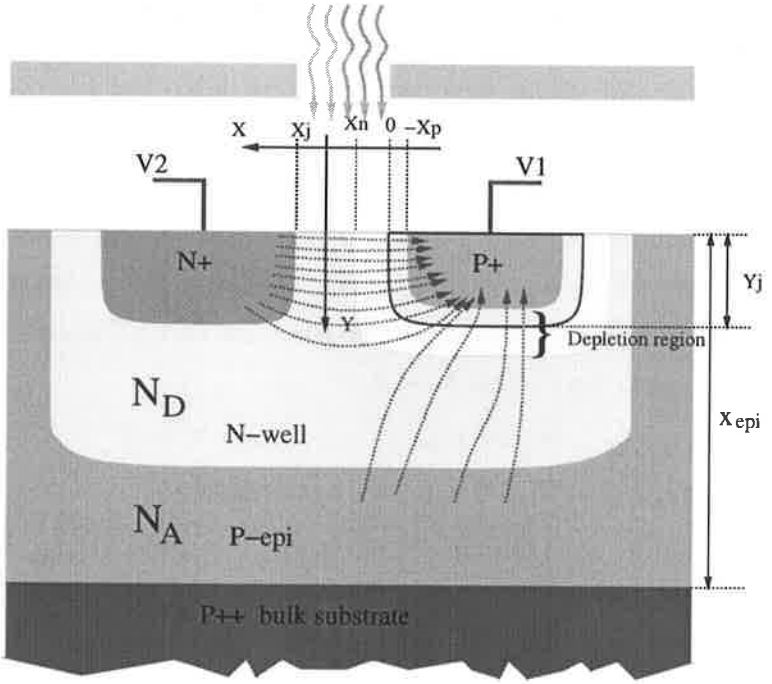


Figure A.4: The structure of a lateral junction diode in an N-Well CMOS process.

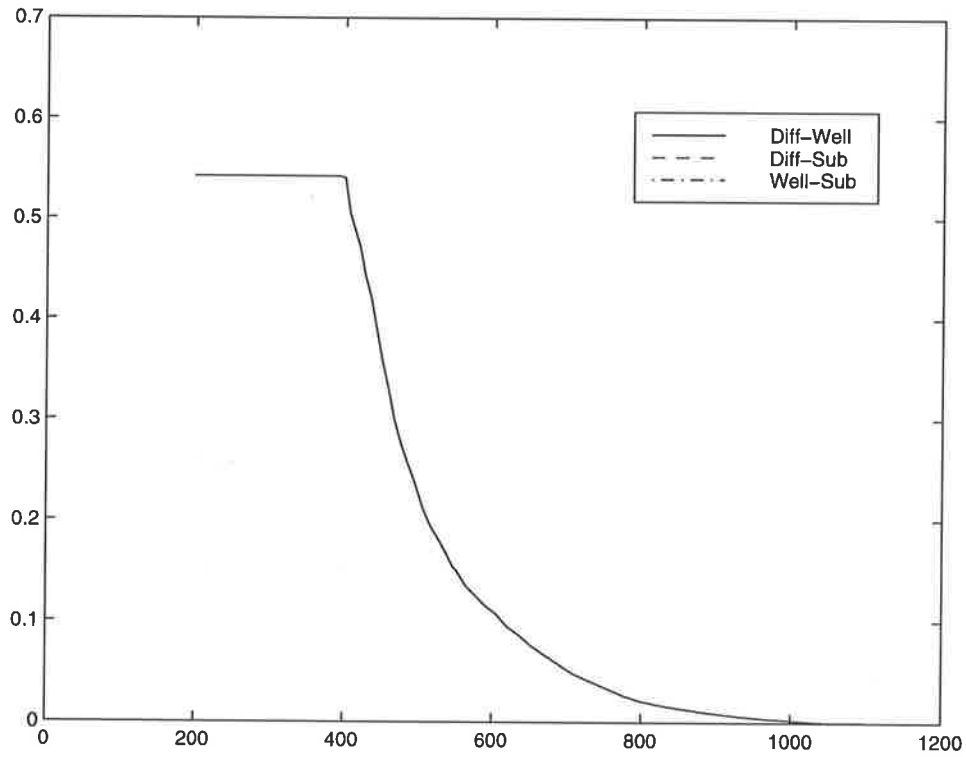


Figure A.5: Simulation result of the lateral photodiode in a 2μm CMOS process.

A.3 Quantum Efficiency of a Vertical Bipolar transistor

The structure of a vertical bipolar transistor is shown in Figure A.6. It is assumed that only the flat area is exposed to light, as otherwise, there will be some contribution from the vertical walls of the emitter-base and base-collector junctions.

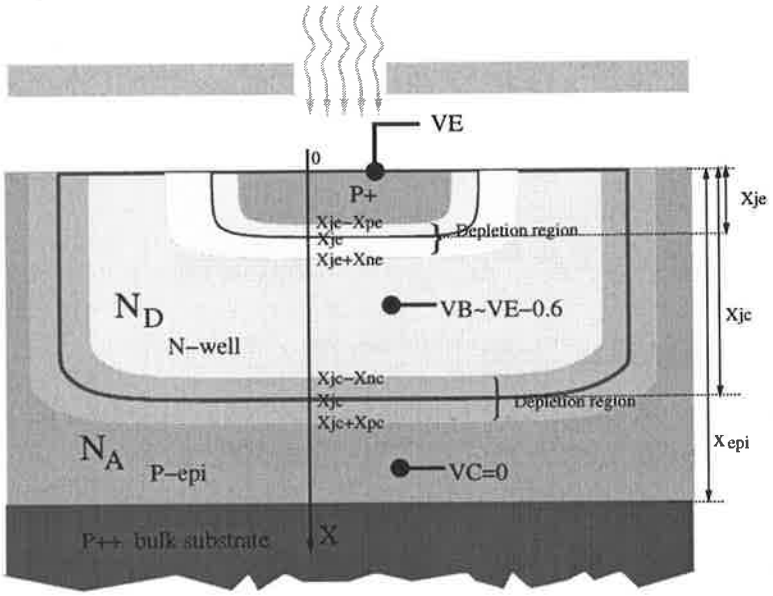


Figure A.6: The structure of a vertical bipolar detector in an N-Well CMOS process.

We can write the diffusion equation in the three regions as:

$$\begin{aligned} D_{ne}\frac{\partial^2 n_{pe}}{\partial x^2} - \frac{n_{pe}-n_{pe0}}{\tau_{ne}} + G(x) = 0 \quad & \text{in the P-Emitter} \\ D_{pb}\frac{\partial^2 p_{nb}}{\partial x^2} - \frac{p_{nb}-p_{nb0}}{\tau_{pb}} + G(x) = 0 \quad & \text{in the N-Base} \\ D_{nc}\frac{\partial^2 n_{pc}}{\partial x^2} - \frac{n_{pc}-n_{pc0}}{\tau_{nc}} + G(x) = 0 \quad & \text{in the P-Collector} \end{aligned} \tag{A.19}$$

The boundary conditions are:

$$\begin{aligned} n_{pe}|_{x=0} &= 0 & n_{pe}|_{x=x_{je}-x_{pe}} &= n_{pe}(e^{\frac{V_{EB}}{V_T}} - 1) & \text{in emitter} \\ p_{nb}|_{x=x_{je}+x_{ne}} &= p_{nb0}(e^{\frac{V_{EB}}{V_T}} - 1) & p_{nb}|_{x=x_{jc}-x_{nc}} &= -p_{nb0} & \text{in base} \\ n_{pc}|_{x=x_{jc}+x_{pc}} &= -n_{pc0} & n_{pc}|_{x=x_{epi}} &= 0 & \text{in collector} \end{aligned} \tag{A.20}$$

The diffusion equations can be solved as follows.

$$
\begin{aligned}
n_{pe}(x) &= n_{pe0} + Ae^{\frac{x}{L_e}} + Be^{-\frac{x}{L_e}} + Ce^{-\alpha x} \\
p_{nb}(x) &= p_{nb0} + Ee^{\frac{x}{L_b}} + Fe^{-\frac{x}{L_b}} + He^{-\alpha x} \\
n_{pc}(x) &= n_{pc0} + Ke^{\frac{x}{L_c}} + Me^{-\frac{x}{L_c}} + Re^{-\alpha x} \\
A &= \frac{-n_{pe0}(e^{\frac{V_{EB}}{V_T}} + e^{-Z} - 2) - C(e^{-Z} - e^{-\alpha(x_{je}-x_{pc})})}{e^{-Z} - e^{+Z}} \\
B &= \frac{-n_{pe0}(e^{\frac{V_{EB}}{V_T}} + e^{+Z} - 2) - C(e^{+Z} - e^{-\alpha(x_{je}-x_{pc})})}{e^{+Z} - e^{-Z}} \\
Z &= \frac{x_{je}-x_{pe}}{L_e} \qquad C = \frac{\Phi_0 \alpha L_e^2}{D_e(1-\alpha^2 L_e^2)} \\
E &= \frac{p_{nb0}(e^{-Y}(e^{\frac{V_{EB}}{V_T}} - 1) + 2e^{-X}) - H(e^{-\alpha(x_{je}+x_{ne})}e^{-Y} - e^{-\alpha(x_{jc}-x_{nc})}e^{-X})}{e^{+X-Y} - e^{-X+Y}} \\
F &= \frac{p_{nb0}(e^{+Y}(e^{\frac{V_{EB}}{V_T}} - 1) + 2e^{+X}) - H(e^{-\alpha(x_{je}+x_{ne})}e^{+Y} - e^{-\alpha(x_{jc}-x_{nc})}e^{+X})}{e^{-X+Y} - e^{+X-Y}}
\end{aligned}
\tag{A.21}
$$

$$
X = \frac{x_{je}+x_{ne}}{L_b} \qquad Y = \frac{x_{jc}-x_{nc}}{L_b} \qquad H = \frac{\Phi_0 \alpha L_b^2}{D_b(1-\alpha^2 L_b^2)}
$$

$$
M = \frac{-n_{pc0}(2e^{+W} - e^{+U}) - R(e^{-\alpha(x_{jc}+x_{pc})}e^{+W} - e^{-\alpha(x_{epi})}e^{-U})}{e^{+W-U} - e^{-W+U}}
$$

$$
K = \frac{-n_{pc0}(2e^{-W} - e^{-U}) - R(e^{-\alpha(x_{jc}+x_{pc})}e^{-W} - e^{-\alpha(x_{epi})}e^{+U})}{e^{-W+U} - e^{+W-U}}
$$

$$
U = \frac{x_{jc}+x_{pc}}{L_c} \qquad W = \frac{x_{epi}}{L_c} \qquad R = \frac{\Phi_0 \alpha L_c^2}{D_c(1-\alpha^2 L_c^2)}
$$

The diffusion component of the emitter and collector currents can be expressed as:

$$
\begin{aligned}
J_{diff,E} &= -qD_e \frac{\partial n_{pe}(x)}{\partial x}\Big|_{x=x_{je}-x_{pe}} + qD_b \frac{\partial p_{nb}(x)}{\partial x}\Big|_{x=x_{je}+x_{ne}} \\
J_{diff,C} &= -qD_c \frac{\partial n_{pc}(x)}{\partial x}\Big|_{x=x_{jc}+x_{pc}} + qD_b \frac{\partial p_{nb}(x)}{\partial x}\Big|_{x=x_{jc}-x_{nc}}
\end{aligned}
\tag{A.22}
$$

The drift components can be simply obtained by integrating the amount of generated electron-hole pairs in the depletion regions.

$$
\begin{aligned}
J_{drift} &= \int_{depletion} -qG(x)dx \\
J_{drift,E} &= -q\Phi_0(e^{-\alpha(x_{je}+x_{ne})} - e^{-\alpha(x_{je}-x_{pe})}) \\
J_{drift,C} &= +q\Phi_0(e^{-\alpha(x_{jc}+x_{pc})} - e^{-\alpha(x_{jc}-x_{nc})})
\end{aligned}
\tag{A.23}
$$

As the base of this device is floating, the collector and emitter currents should be equal. The only variable parameter, which is unknown, is V_{EB}. The value of V_{EB} for which $I_C = I_E$ can be found using numerical methods. Figure A.7 shows the quantum efficiency of a typical parasitic PNP transistor in a 2μm process. The large gain is simply due to the current gain of the bipolar transistor, which is larger than one. Simulations reveal that the current gain is highly dependent on the base and emitter doping densities. As expected the response is relatively flat over the visible spectrum, which is due to the presence of two junctions at two different depths in the device.

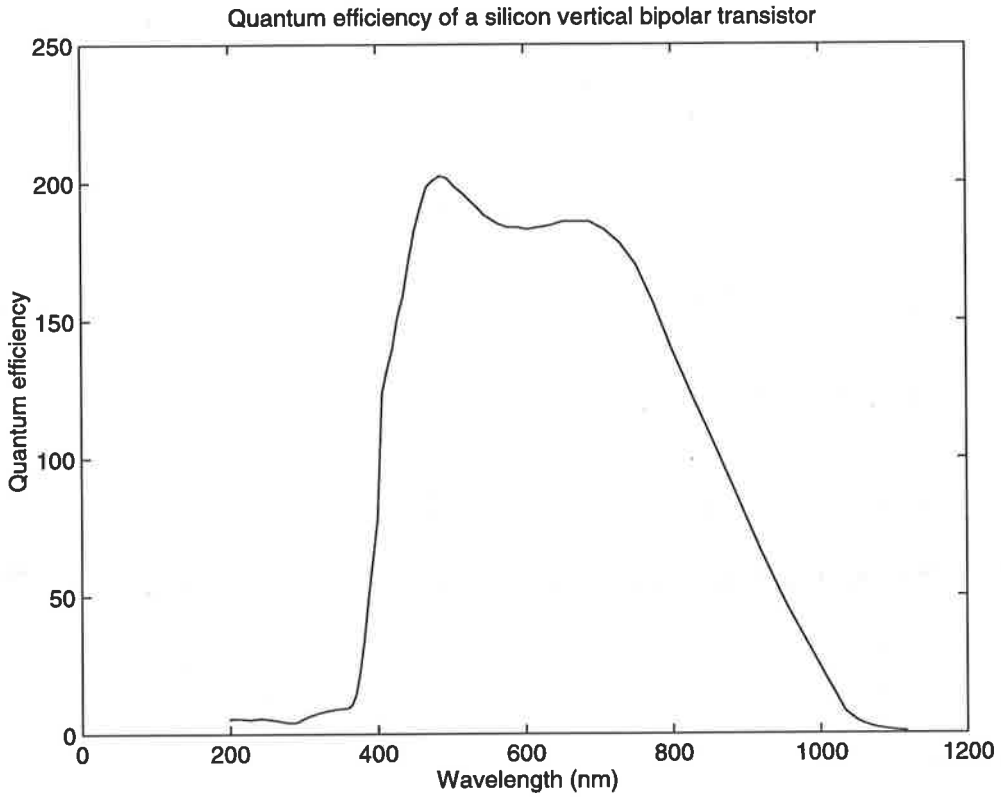


Figure A.7: Simulated quantum efficiency of a vertical bipolar transistor in a 2μm CMOS process. Note that the quantum efficiency is greater than "1", due to the current gain of the transistor.

A.4 Quantum Efficiency of a Lateral Bipolar Photodetector

The structure of a lateral bipolar device is shown in Figure A.8. The simplifying assumptions are similar to those made in Section A.2, except that exposure to light is assumed to be only in the area between the emitter and collector diffusions, and the depletion regions.

Note that in Equation A.24, x denotes the horizontal axis and y the vertical axis. Also $y_j$is the depth of the collector/emitter junctions.

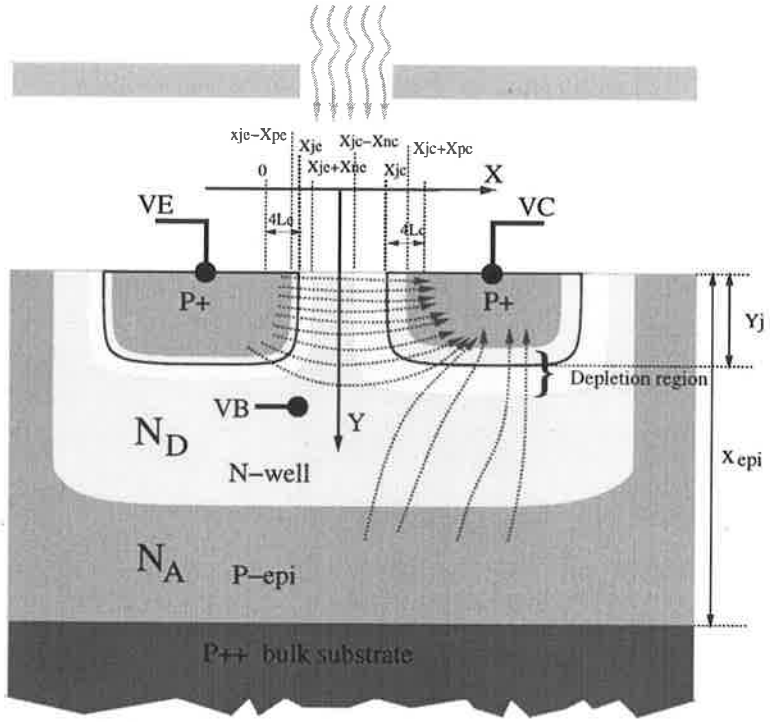


Figure A.8: The structure of a lateral bipolar detector in an N-Well CMOS process.

The diffusion equations in the three regions can be written as:

$$\begin{aligned} D_{ne}\frac{\partial^2 n_{pe}}{\partial x^2} - \frac{n_{pe}-n_{pe0}}{\tau_{ne}} &= 0 && \text{in the P-Emitter} \\ D_{pb}\frac{\partial^2 p_{nb}}{\partial x^2} - \frac{p_{nb}-p_{nb0}}{\tau_{pb}} + G(y) &= 0 && \text{in the N-Base} \\ D_{nc}\frac{\partial^2 n_{pc}}{\partial x^2} - \frac{n_{pc}-n_{pc0}}{\tau_{nc}} &= 0 && \text{in the P-Collector} \end{aligned} \tag{A.24}$$

The diffusion length in the collector and emitter regions is very short. Therefore, we make another simplifying assumption, in that these junctions extend to four times the diffusion length in these regions. We set the origin at $4L_e$ before the start of the emitter junction, to be able to reuse the derivations for the vertical bipolar transistor.

The boundary conditions for the three regions are:

$$\begin{aligned} n_{pe}|_{x=0} &= 0 & n_{pe}|_{x=x_{je}-x_{pe}} &= n_{pe}(e^{\frac{V_{EB}}{V_T}} - 1) && \text{in emitter} \\ p_{nb}|_{x=x_{je}+x_{ne}} &= p_{nb0}(e^{\frac{V_{EB}}{V_T}} - 1) & p_{nb}|_{x=x_{jc}-x_{nc}} &= -p_{nb0} && \text{in base} \\ n_{pc}|_{x=x_{jc}+x_{pc}} &= -n_{pc0} & n_{pc}|_{x=x_{jc}+4L_c} &= 0 && \text{in collector} \end{aligned} \tag{A.25}$$

$$
\begin{aligned}
n_{pe}(x) &= n_{pe0} + Ae^{\frac{x}{L_e}} + Be^{-\frac{x}{L_e}} \\
p_{nb}(x) &= p_{nb0} + G(y)\tau_b + Ee^{\frac{x}{L_b}} + Fe^{-\frac{x}{L_b}} \\
n_{pc}(x) &= n_{pc0} + Ke^{\frac{x}{L_c}} + Me^{-\frac{x}{L_c}} \\
A &= \frac{-n_{pe0}(e^{\frac{V_{EB}}{V_T}} + e^{-Z} - 2)}{e^{-Z} - e^{+Z}} \\
B &= \frac{-n_{pe0}(e^{\frac{V_{EB}}{V_T}} + e^{+Z} - 2)}{e^{+Z} - e^{-Z}} \\
Z &= \frac{x_{je} - x_{pe}}{L_e} \qquad x_{je} = 4L_e \\
E &= (p_{nb0} + G(y)\tau_{pb})E_1 = (p_{nb0} + G(y)\tau_{pb})\frac{(e^{-Y}(e^{\frac{V_{EB}}{V_T}} - 1) + 2e^{-X})}{e^{+X-Y} - e^{-X+Y}} \\
F &= (p_{nb0} + G(y)\tau_{pb})F_1 = (p_{nb0} + G(y)\tau_{pb})\frac{(e^{+Y}(e^{\frac{V_{EB}}{V_T}} - 1) + 2e^{+X})}{e^{-X+Y} - e^{+X-Y}} \\
X &= \frac{x_{je} + x_{ne}}{L_b} \qquad Y = \frac{x_{jc} - x_{nc}}{L_b} \\
U &= \frac{x_{jc} + x_{pc}}{L_c} \qquad W = \frac{x_{jc} + 4L_c}{L_c} \\
M &= \frac{-n_{pc0}(2e^{+W} - e^{+U})}{e^{+W-U} - e^{-W+U}} \qquad K = \frac{-n_{pc0}(2e^{-W} - e^{-U})}{e^{-W+U} - e^{+W-U}}
\end{aligned}
\tag{A.26}
$$

The diffusion currents at the collector and emitter are:

$$
\begin{aligned}
J_{diff,E}(y) &= -qD_e\frac{\partial n_{pe}(x)}{\partial x}|_{x=x_{je}-x_{pe}} + qD_b\frac{\partial p_{nb}(x)}{\partial x}|_{x=x_{je}+x_{ne}} \\
J_{diff,C}(y) &= -qD_c\frac{\partial n_{pc}(x)}{\partial x}|_{x=x_{jc}+x_{pc}} + qD_b\frac{\partial p_{nb}(x)}{\partial x}|_{x=x_{jc}-x_{nc}}
\end{aligned}
\tag{A.27}
$$

The drift currents in the depletion regions of the collector and emitter junctions are:

$$
\begin{aligned}
J_{drift,E}(y) &= -qG(y)x_{ne} \\
J_{drift,C}(y) &= -qG(y)x_{nc}
\end{aligned}
\tag{A.28}
$$

The emitter and collector currents can be obtained by integrating the corresponding drift and diffusion components of each current over the range $[y = 0 \text{ to } y = y_j]$. Notice that the current density is per unit width of the device, and hence it should be divided by the junction depth y_j to yield a current density per unit area. The simulation result for a PNP device with minimum diffusion spacing (3λ) in a 2μm CMOS process is shown in Figure A.9. The general shape of the quantum efficiency is very similar to that of a lateral photodiode (Figure A.5).

$$
\begin{aligned}
J_{drift,E} &= -q\Phi_0 x_{ne}(1-e^{-\alpha y_j}) \\
J_{drift,C} &= -q\Phi_0 x_{nc}(1-e^{-\alpha y_j})
\end{aligned}
$$

$$
\begin{aligned}
J_{diff,E} &= \tfrac{1}{y_j}\int_{y=0}^{y=y_j} j_{diff,E}(y)dy \\
&= -qD_e\{\tfrac{A}{L_e}e^{Z} - \tfrac{B}{L_e}e^{-Z}\} + qD_b p_{nb0}\{\tfrac{E_1}{L_b}e^{X} - \tfrac{F_1}{L_b}e^{-X}\} \\
&\quad +qD_b\tau_{pb}p_{nb0}\{\tfrac{E_1}{L_b}e^{X} - \tfrac{F_1}{L_b}e^{-X}\}\{\tfrac{\Phi_0}{y_j}(1-e^{-\alpha y_j})\}
\end{aligned}
\qquad \text{(A.29)}
$$

$$
\begin{aligned}
J_{diff,C} &= \tfrac{1}{y_j}\int_{y=0}^{y=y_j} j_{diff,C}(y)dy \\
&= -qD_c\{\tfrac{K}{L_c}e^{U} - \tfrac{M}{L_c}e^{-U}\} + qD_b p_{nb0}\{\tfrac{E_1}{L_b}e^{Y} - \tfrac{F_1}{L_b}e^{-Y}\} \\
&\quad +qD_b\tau_{pb}p_{nb0}\{\tfrac{E_1}{L_b}e^{Y} - \tfrac{F_1}{L_b}e^{-Y}\}\{\tfrac{\Phi_0}{y_j}(1-e^{-\alpha y_j})\}
\end{aligned}
$$

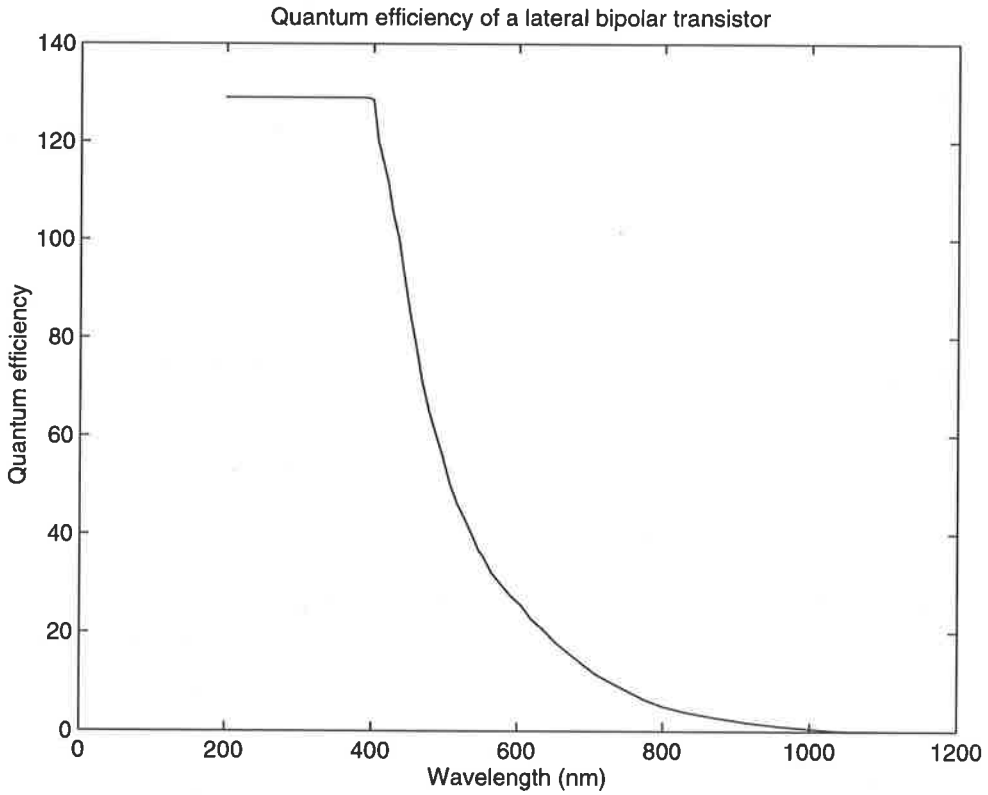


Figure A.9: Simulated quantum efficiency of a vertical bipolar transistor in a 2μm CMOS process.

A.5 Mixed structures

The simulation results for the lateral and vertical devices obtained in the previous sections indicate that vertical devices have a relatively flat response over the visual spectrum, while lateral devices have a better blue response. The lateral and vertical devices can be combined in simple fashion to form new structures. For the photodiode structures all that is required is to make the exposure window opening large enough for the edges of the diode to be exposed. Figure A.10 illustrates the mixed devices.

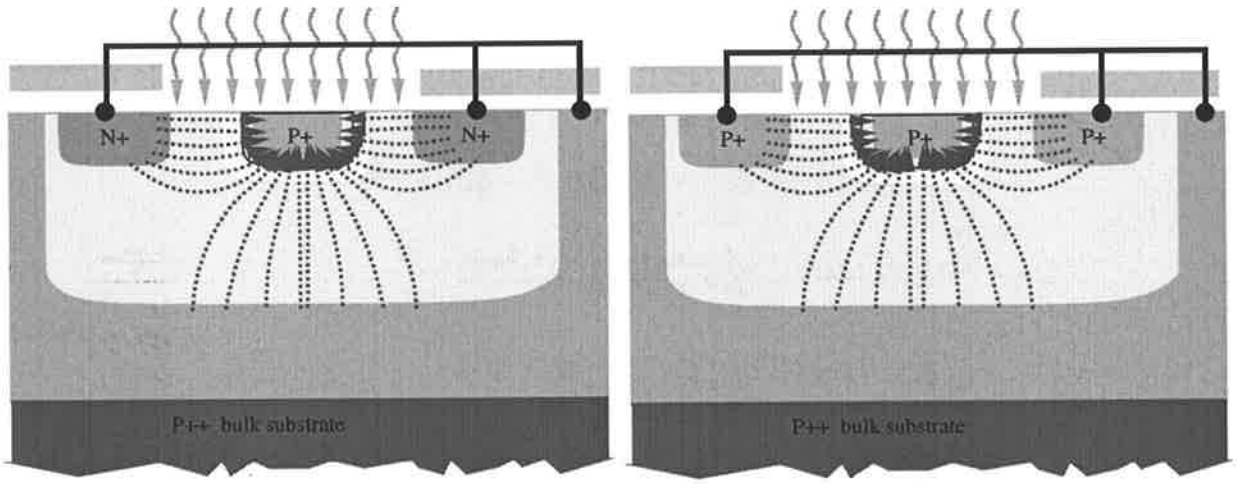


Figure A.10: a) A mixed lateral and vertical photodiode. b) A mixed lateral and vertical bipolar transistor.

A.6 Quantum Efficiency of a Photogate

The structure of a photogate is shown in Figure A.11. A photogate is nothing but a MOS capacitor exposed to light. A photogate operates by integrating the photogenerated carriers in the potential well, which is created by applying a large voltage to the gate. A simple assumption made here is that the depth of the depletion region is small. One can verify this using the following equation.

$$\begin{aligned} x_d &= \frac{-B+\sqrt{B^2+4AV_G}}{2A} \\ A &= \frac{qN_S}{2\epsilon_{si}\epsilon_0} \\ B &= \frac{qN_S T_{ox}}{2\epsilon_{sio2}\epsilon_0} \end{aligned} \tag{A.30}$$

In a 2μm process the typical values for x_d are less than 0.5μm. Therefore, it is reasonable to assume that all the charges filling the potential well are diffusing from areas outside the

depletion region.

One important drawback of photogates is that they have very poor blue response because the gate material absorbs this part of the spectrum. In new processes the gate is silicided, which blocks most parts of the visual spectrum, and hence the silicide layer should be masked out from the areas above the photogate. Another solution is to make several windows in the gate so that light can pass through. Even with polysilicon gates it is recommended to use windowed gates for the photogate devices.

The spectral response of the photogate is obtained by solving the diffusion equation in the substrate area. Notice that a photogate works in a *reset-and-integrate* mode. During the reset cycle, charges are emptied from the potential well, and during the integration cycle, diffusion of photogenerated currents fills up the potential well.

$$
\begin{aligned}
J_{diff} &= -\frac{qD_s}{L_s}Ae^{X} + \frac{qD_s}{L_s}Be^{-X} - qC\alpha e^{-\alpha x_d} \\
A &= \frac{-n_{ps0}e^{-Y} - C(e^{-X}e^{-\alpha x_{epi}} - e^{-Y}e^{-\alpha x_d})}{e^{+X-Y} - e^{-X+Y}} \\
B &= \frac{-n_{ps0}e^{+Y} - C(e^{+X}e^{-\alpha x_{epi}} - e^{+Y}e^{-\alpha x_d})}{e^{-X+Y} - e^{+X-Y}} \\
X &= \frac{x_d}{L_s} \qquad Y = \frac{x_{epi}}{L_s} \qquad C = \frac{\Phi_1 \alpha L_e^2}{D_e(1-\alpha^2 L_e^2)}
\end{aligned}
\tag{A.31}
$$

Φ_1 is the photon flux at the surface of the silicon. If we assume that the gate material is polysilicon and has the same absorption coefficient as silicon, we will have:

$$
\Phi_1 = \Phi_0 e^{-\alpha T_{gate}} \tag{A.32}
$$

The simulated spectral response of the photogate is shown in Figure A.12. The device has a better response for the red part of the spectrum, and the blue response is significantly lower, when the effect of the absorption of the gate is considered.

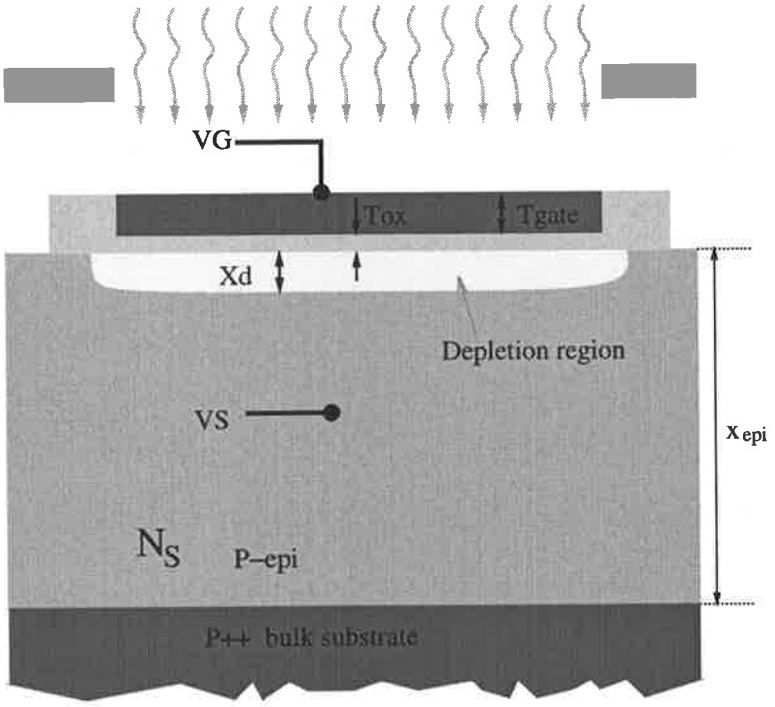


Figure A.11: Structure of a photogate device in an N-Well CMOS process.

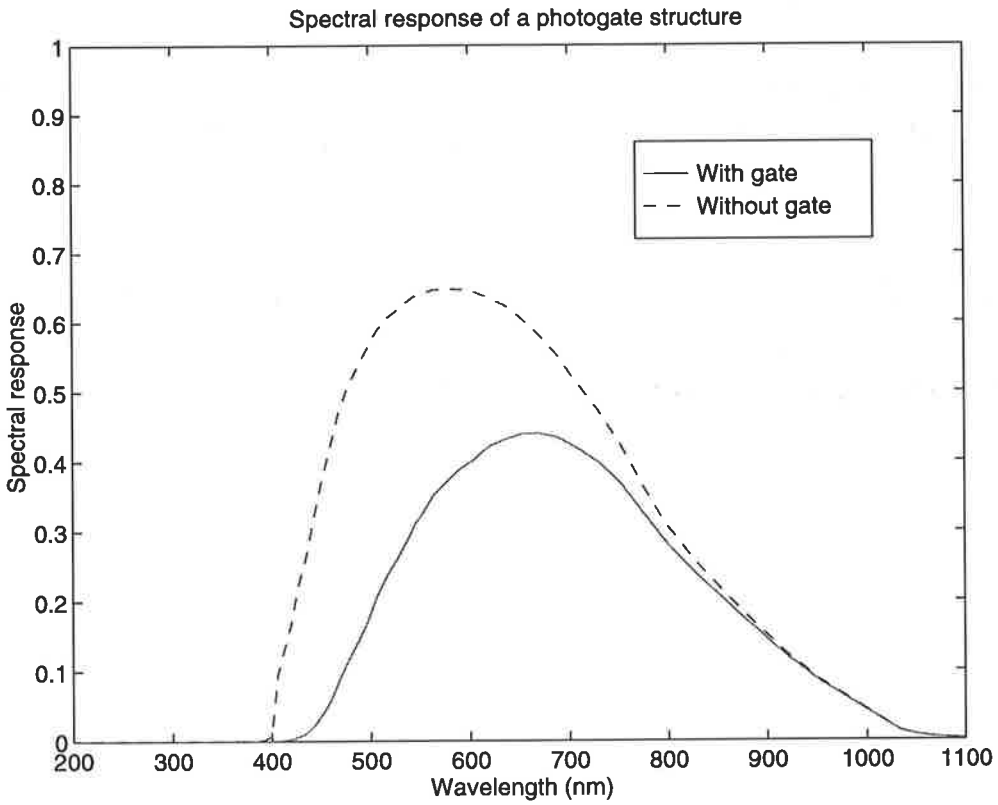


Figure A.12: Spectral response of the photogate showing the effect of the gate absorption in the reduction of quantum efficiency.

Appendix B

Analysis of Second-Order Resistive Network

B.1 Stability

The stability criteria for a general Nth order network has been derived elegantly by [Matsumoto et al. 93]. However, only the conditions under which the network is stable are addressed, and hence we present here an alternative method based on the analysis of the poles of the network. This method, in addition to giving conditions for stability, yields the location of the poles, and therefore can be used in determining the shape of the convolution kernel of the specific network. Also this method is more suited for a computer synthesis of a resistive network with the desired kernel function.

For the second-order resistive network shown in Figure B.1 the Z-transform can be written as:

$$
\begin{aligned}
x(n) &= -g_2 y(n-2) - g_1 y(n-1) + (g_0 + 2g_1 + 2g_2) y(n) - g_1 y(n+1) - g_2 y(n+2) \\
H(z) &= \frac{-\frac{1}{v} z^2}{z^4 + \frac{u}{v} z^3 - \frac{1+2u+2v}{v} z^2 + \frac{u}{v} z + 1} \\
u &= \frac{g_1}{g_0} \quad v = \frac{g_2}{g_0}
\end{aligned}
\tag{B.1}
$$

I have chosen u and v as variables because they represent a more physically comprehensible space for understanding the effect of the resistor values on stability. The poles of the system can be easily found.

$$
\begin{aligned}
p_1, p_2, p_3, p_4 &= \frac{-A \pm \sqrt{A^2 - 4}}{2} \\
A &= \frac{u \pm \sqrt{(u+4v)^2 + 4v}}{2v} \\
p_1 p_2 &= 1 \text{ and } p_3 p_4 = 1
\end{aligned}
\tag{B.2}
$$

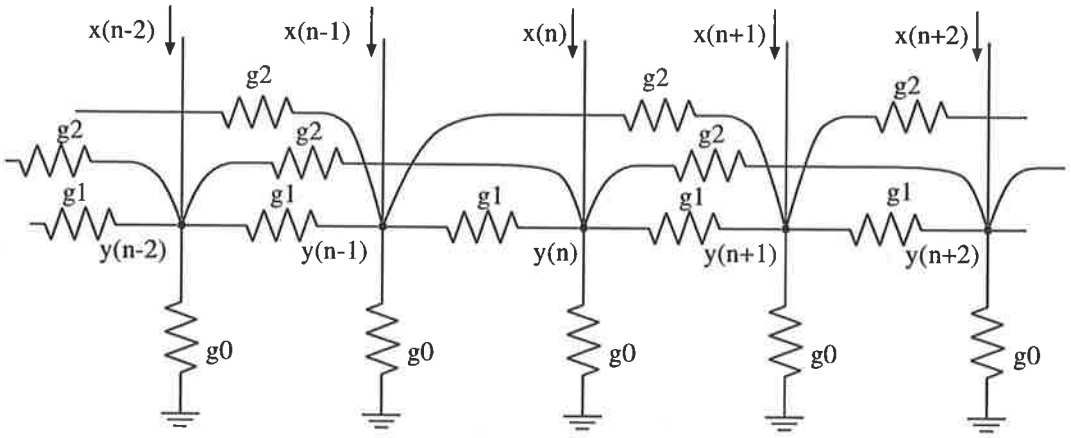


Figure B.1: Second-order resistive network.

This system is *noncausal* and therefore in general it may not have a unique impulse response, if sufficient constraints are not applied to the system. In causal systems the constraint that the input and output sequence are "zero" for all left-values (or past values for time sequences), guarantees a unique impulse response for an LTI (linear time-invariant) system. In our case, which is a spatial linear noncausal system, the additional constraint is that the system should be symmetrical. In other words the poles of the system are present in pairs such that $p_{n1}p_{n2} = 1$. In fact from the Z-transform of a general Nth order network given by

$$H(z) = \frac{1}{\sum_{k=-N}^{+N} a_k z^k} \qquad (B.3)$$

$$a_{-n} = a_n = -g_n \text{ for } n \neq 0 \qquad a_0 = g_0 + 2\sum_{1}^{n} g_n$$

it can be seen that if p_{n1} is a pole of the system, $p_{n2} = 1/p_{n1}$ would be another pole of the system.

Under these conditions the system would be unstable when the poles of the system are on the unit circle in the Z domain. For the second-order network the poles derived in Equation B.2 can be analyzed as follows.

1. $u > 0, v > 0$ (Region 1)

   In this case A would be real and

$$A^2 - 4 = \frac{2u^2 + 8uv + 4v \pm 2u\sqrt{(u+4v)^2 + 4v}}{4v^2} > 0 \qquad (B.4)$$

   It can be seen that all the poles are real and therefore in this region the network is stable, and the impulse response will consist of two exponential decay functions.

2. $u < 0, v > 0$ (Region 2)

In this case still A is real. However, by checking the expression $A^2 - 4$ one can see that

$$(2u^2 + 8uv + 4v)^2 < 4u^2((u + 4v)^2 + 4v) \quad \Rightarrow u < -\frac{1}{4} \tag{B.5}$$

or $A^2 - 4 < 0$. The system will have two imaginary and two real poles. The magnitude of the imaginary poles will be

$$\sqrt{\frac{A^2 + \left(\sqrt{|A^2 - 4|}\right)^2}{4}} = 1$$

Hence the network would be unstable for all $u < -\frac{1}{4}$.

3. $v < 0$ and $(u + 4v)^2 + 4v < 0$ (Region 3)

In this case A will be complex, and $\sqrt{A^2 - 4}$ will also be a complex number.

$$\begin{aligned}\sqrt{A^2 - 4} &= \sqrt{\overbrace{\frac{2u^2 + 8uv + 4v}{4v^2}}^{X} \pm i\overbrace{\frac{2u\sqrt{|(u + 4v)^2 + 4v|}}{4v^2}}^{Y}} \\ &= \sqrt{X \pm iY} \\ &= \pm\sqrt{\frac{X + \sqrt{X^2 + Y^2}}{2}} + i\sqrt{\frac{-X + \sqrt{X^2 + Y^2}}{2}}\end{aligned} \tag{B.6}$$

Therefore, the poles of the network will be

$$p_n = \frac{-A \pm \left(\pm\sqrt{\frac{X + \sqrt{X^2 + Y^2}}{2}} + i\sqrt{\frac{-X + \sqrt{X^2 + Y^2}}{2}}\right)}{2} \tag{B.7}$$

Note that the first and the second $\pm$ sign are independent. The magnitude of the poles after some simplification will be

$$|p_n|^2 = \frac{A^2 + \sqrt{X^2 + Y^2} \pm A\sqrt{2(X + \sqrt{X^2 + Y^2})}}{4} \tag{B.8}$$

There are no values for which the magnitude of poles becomes "1", and therefore the network is stable in this region. The impulse response will consist of two exponentially modulated sinusoidal signals.

It is worth noting that $(u + 4v)^2 + 4v = 0$ represents a parabola which lies in the quarter-plane $v < 0, u > -\frac{1}{4}$.

4. $v < 0$, $(u + 4v)^2 + 4v > 0$ and $u < -\frac{1}{4}$ (Regions 4, 5, and 6)

This region is a part of the quarter-plane $[v < 0, u > -\frac{1}{4}]$ which is outside the parabola

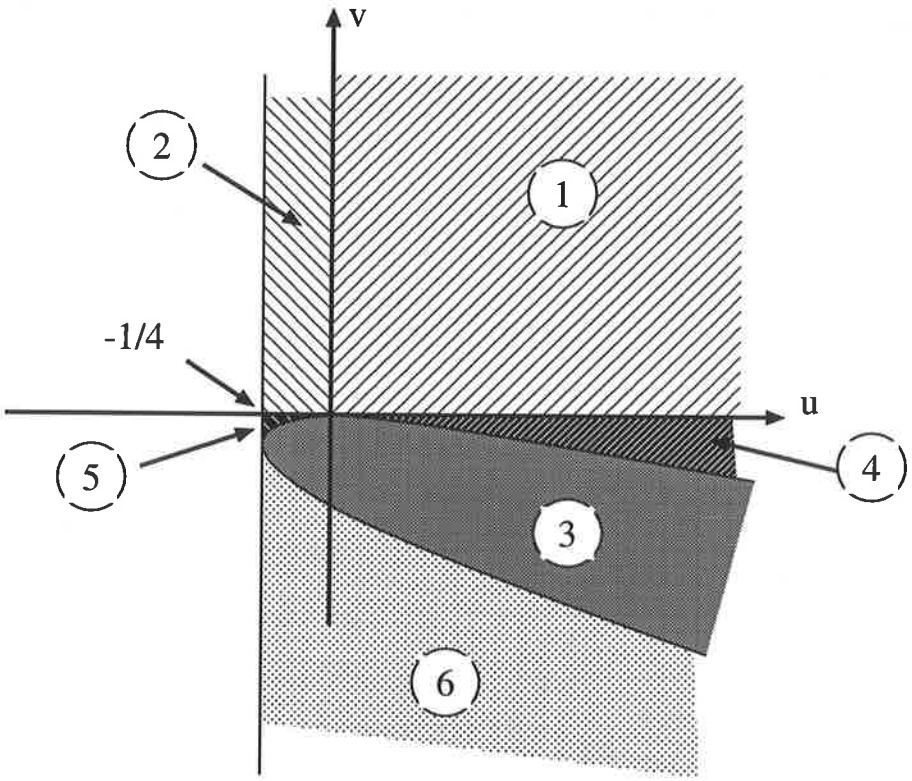


Figure B.2: Stability regions of the second-order resistive network.

$(u+4v)^2+4v=0$. Three regions, 4,5, and 6, can be recognized. We have

$$(u+4v)^2+4v=0 \quad \Rightarrow \begin{cases} v=\dfrac{-(2u+1)\pm\sqrt{4u+1}}{8} \\ u=-4v\pm\sqrt{-4v} \end{cases} \tag{B.9}$$

Region 4 is characterized by $0>v>\dfrac{-(2u+1)+\sqrt{4u+1}}{8}, u>-4v+\sqrt{-4v}>0$, region 5 by $0>v>\dfrac{-(2u+1)+\sqrt{4u+1}}{8}, -1/4<u<-4v-\sqrt{-4v}$, and region 6 by $v<\dfrac{-(2u+1)-\sqrt{4u+1}}{8}, u>-1/4$. In all these regions, A is real, and the stability depends on the sign of A^2-4. If the sign is negative, the network will have poles on the unit circle and therefore the network becomes unstable. Otherwise the network will be stable.

$$A^2-4>0 \;\Rightarrow\; (A-2)(A+2)>0 \tag{B.10}$$

Note that $v<0$. When $u>0$ (in region 4) we have

$$\begin{aligned} &u+4v>\sqrt{-4v}>0 \;\Rightarrow\; (u+4v)>\sqrt{(u+4v)^2+4v} \;\Rightarrow \\ &\frac{(u+4v)-\sqrt{(u+4v)^2+4v}}{2v}<0 \end{aligned} \tag{B.11}$$

This means that $A+2<0$, and therefore A^2-4 is positive in this region. As a result the network is stable.

In region 5, $u<0$, and it is rather easy to see that we have

$$\begin{aligned} &u-4v<0 \;\Rightarrow\; u-4v-sqrt(u+4v)^2+4v<0 \;\Rightarrow \\ &\frac{(u-4v)-\sqrt{(u+4v)^2+4v}}{2v}>0 \end{aligned} \tag{B.12}$$

Therefore $A-2>0$, which means that A^2-4 is positive in this region. Hence all the poles of the system are real and the system is stable.

In the region below the parabola (region 6) we prove that $A^2-4<0$. For this purpose we show that $A-2<0$ and $A+2>0$. In order to have $A-2<0$, we should have $u-4v\pm\sqrt{(u+4v)^2+4v}>0$. This can easily be shown as in this region we have $u-4v>0$. Therefore we obtain

$$u-4v-\sqrt{(u+4v)^2+4v}>0 \Leftarrow (u-4v)^2>(u+4v)^2+4v \Leftarrow u<-\tfrac{1}{4} \tag{B.13}$$

To show that $A+2>0$, $u+4v\pm\sqrt{(u+4v)^2+4v}<0$, or $(u+4v)^2-[(u+4v)^2+4v]>0$. Which is true in this region. Therefore, we have proven that $A^2-4<0$. Hence all the poles of the system lie on the unit circle, and the network is unstable.

In general, the poles of an Nth order network can be found using numerical techniques, and the stability can be checked by looking at the position of the poles with respect to the unit circle.

B.2 Impulse Response

The impulse response of the resistive network can be found relatively easily, once the poles of the network have been obtained from Equation B.2. It should be noted that although the system has four poles, only two poles determine the impulse response in the stable regions. This is because the system is noncausal, and we have applied the constraint that the impulse response is symmetrical. There are two pairs of poles such that

$$\begin{aligned} p_1p_2&=1 \\ p_3p_4&=1 \end{aligned} \tag{B.14}$$

One of the poles in each pair is associated with the left-hand impulse response, and the other one with the right-hand response. In the stable region, as the condition in Equation B.14 holds, we only need to consider the poles with a magnitude less that 1, in order to determine the impulse

response. The transfer function obtained in Equation B.1 can be rewritten as

$$
\begin{aligned}
H(z) &= \frac{-\frac{1}{v}z^2}{z^4 + \frac{u}{v}z^3 - \frac{1+2u+2v}{v}z^2 + \frac{u}{v}z + 1} \\
&= -\frac{1}{v}z^2 \left\{ \frac{\frac{1}{A_1 - A_2}z + \frac{A_1}{A_1 - A_2}}{z^2 + A_1 + 1} + \frac{\frac{-1}{A_1 - A_2}z + \frac{-A_2}{A_1 - A_2}}{z^2 + A_2 + 1} \right\} \\
&= -\frac{1}{v}z^2 \left\{ \left[\frac{-\frac{A_1 + p_1}{(p_2 - p_1)(A_1 - A_2)}}{z - p_1} + \frac{\frac{A_1 + p_2}{(p_2 - p_1)(A_1 - A_2)}}{z - p_2} \right] \right. \\
&\quad \left. + \left[\frac{\frac{A_2 + p_3}{(p_4 - p_3)(A_1 - A_2)}}{z - p_3} + \frac{-\frac{A_2 + p_4}{(p_4 - p_3)(A_1 - A_2)}}{z - p_4} \right] \right\}
\end{aligned}
\tag{B.15}
$$

where

$$
\begin{aligned}
A_1, A_2 &= \frac{u \pm \sqrt{(u+4v)^2 + 4v}}{2v} \\
p_1, p_2 &= \frac{-A_1 \mp \sqrt{A_1^2 - 4}}{2} \\
p_3, p_4 &= \frac{-A_2 \mp \sqrt{A_2^2 - 4}}{2}
\end{aligned}
\tag{B.16}
$$

1. **Regions 1 and 2:**

   In these regions A_1 and A_2 are both real. It is easy to see that one pair of poles (p_1, p_2) are alway negative and the other always positive. Note that both poles in a pair have the same sign.

   What is more interesting, is the relative magnitude of the poles. In region 1 the magnitude of the positive pole is larger than that of the negative pole, and vice versa in region 2. On the Y-axis, where $u = 0$, both poles have the same magnitude but opposite signs. The insets a, b, c, d, and e in Figure B.3, clearly show the manner in which the kernel function changes in regions 1 and 2. The oscillating component comes from the negative pole.

2. **Region 3:**

   In this region the poles are complex, and as we derived in Equation B.7

$$
\begin{aligned}
p_2, p_1 &= \frac{-A_1 \pm \left(+\sqrt{\frac{X + \sqrt{X^2 + Y^2}}{2}} + i\sqrt{\frac{-X + \sqrt{X^2 + Y^2}}{2}} \right)}{2} \\
p_4, p_3 &= \frac{-A_2 \pm \left(-\sqrt{\frac{X + \sqrt{X^2 + Y^2}}{2}} + i\sqrt{\frac{-X + \sqrt{X^2 + Y^2}}{2}} \right)}{2} \\
A_1, A_2 &= \frac{u \pm i\sqrt{|(u+4v)^2 + 4v|}}{2v}
\end{aligned}
\tag{B.17}
$$

It can be seen that $p_1 = p_4^*$, and $p_2 = p_3^*$. Therefore, if for example p_1 is the pole within the unit circle, p_4 will also be inside the unit circle. The shape of the kernel function is an exponentially modulated sinusoid.

3. **Regions 4 and 5:**

In region 4 both poles are real and positive. However, the coefficient of one of the exponential term of the impulse response, associated with one of the poles, is negative, while the other is positive. Hence, the kernel function is the difference of two exponentials (DoE). The DoE has been used in several vision chips to approximate the difference of Gaussians (DoG).

In region 5 both poles are real and negative. Again the coefficient of one of the exponential terms in the impulse response is positive and the other one is negative.

Figure B.3 illustrates some of the kernel functions in different regions of stability. Apart from plot (a) in the figure, which lies on the boundary of the stability region, the rest of the curves belong to points within the stability regions. The plots have been obtained by simulating a 127-cell resistive network with the input to all cells being zero, except the middle cell which has an input of "1".

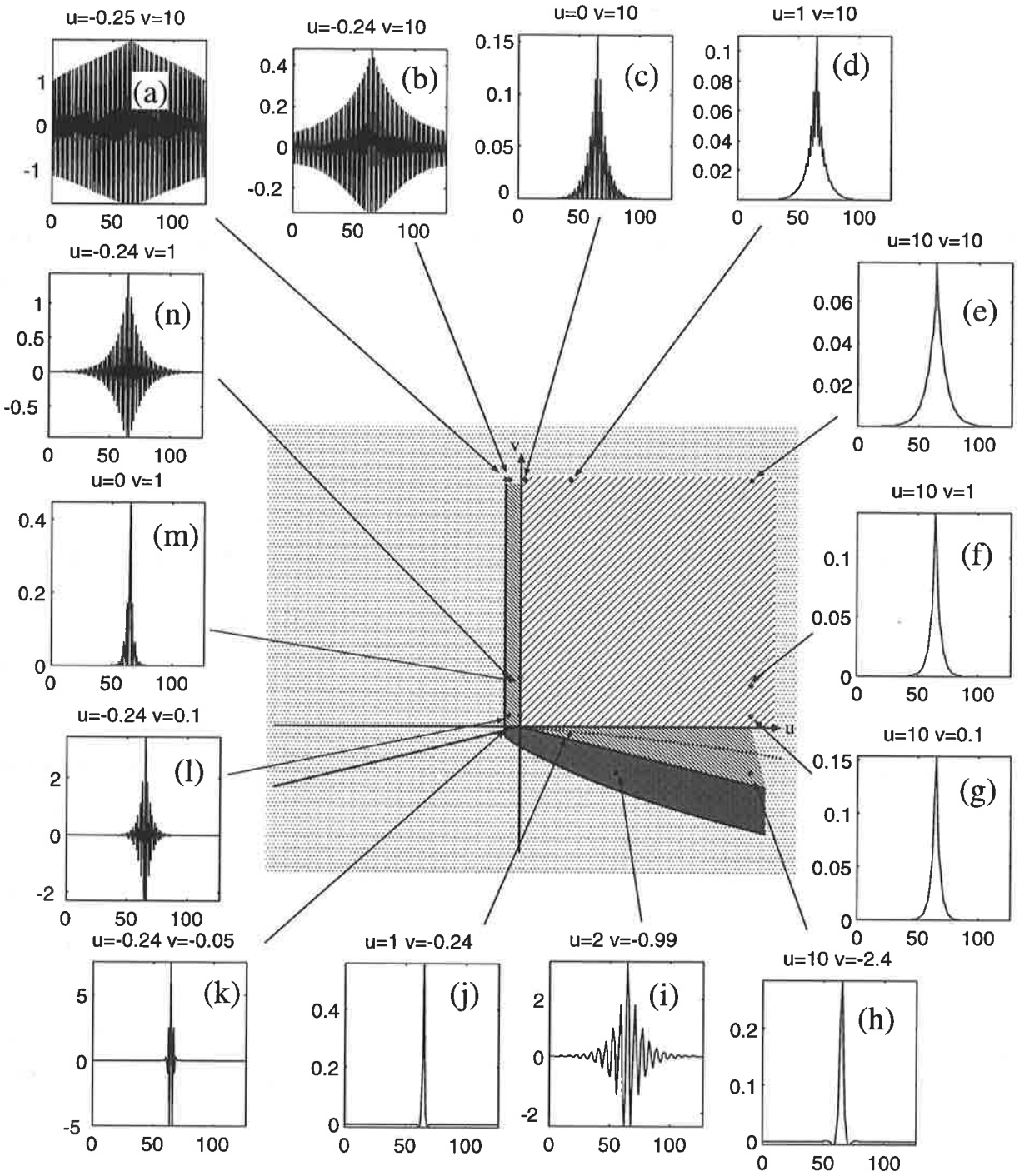


Figure B.3: Kernel functions of the second-order resistive network for several sample points. Notice the values on the vertical axis of the plots.

Bibliography

[Adelson and Bergen 85] E.H. Adelson & J. R. Bergen, "Spatio-temporal energy models for the perception of motion," *Journal of Optica Society of America*, Vol. A2, pp. 284–299, 1985.

[Ahumada Jr. 91] A.J. Ahumada Jr., "Learning receptor positions," In M.S. Landy & J.A. Movshon, editeurs, Computational Models of Visual Processing, chapter 2, pp. 23–34, The MIT Press, Cambridge, Massachusetts, 1991.

[Aizawa et al. 94] K. Aizawa, H. Ohno, T. Hamamoto, M. Hatori & J. Yamazaki, "A novel image sensor for video compression," *Proceedings ICIP*, Vol. 3, pp. 591–595, 1994.

[Aizawa et al. 95] K. Aizawa, Y. Egi, T. Hamamoto, M. Hatori & J. Yamazaki, "An image sensor for on-sensor-compression," *Workshop on Computer Architecture for Machine Perception*, pp. 14–20, 1995.

[Aizawa et al. 97] K. Aizawa, Y. Egi, T. Hamamoto, M. Hatori, M. Abe, H. Maruyama & H. Otake, "Computational image sensor for on sensor compression," *IEEE Trans. Electron Devices*, Vol. 44, No. 10, pp. 1724–1730, October 1997.

[Andreou and Boahen 94] A.G. Andreou & K.A. Boahen, "Neural information processing II," In M. Ismail & T. Fiez, editeurs, Analog VLSI signal and information processing, chapter 8, pp. 358–413, McGraw-Hill, 1994.

[Andreou and Boahen 95] A.G. Andreou & K.A. Boahen, "A 590,000 transistor 48,000 pixel, contrast sensitive, edge enhancing, CMOS imager-silicon retina," *in Proc. 16th Conf. Advanced Research in VLSI*, pp. 225–240, 1995.

[Andreou and Boahen 96] A.G. Andreou & K.A. Boahen, "Translinear circuits in subthreshold MOS," *Analog Integrated Circuits and Signal Processing*, Vol. 9, No. 2, pp. 141–166, March 1996.

[Andreou et al. 91] A.G. Andreou, K.A. Boahen, P.O. Pouliquen, A. Pavasovic, R.E. Jenkins & K. Strohbehn, "Current-mode subthreshold MOS circuits for analog VLSI neural systems," *IEEE Trans. Neural Networks*, Vol. 2, No. 2, pp. 205–213, March 1991.

[Åström 93] Andres Åström. *Smart image sensors.* PhD thesis, Department of Electrical Engineering, Likoping University, S-581 83 Linkoping, Sweden, 1993.

[Barlow and Lewick 65] H.B. Barlow & W.R. Lewick, "The mechanism of directionally selective units in the rabbit's retina," *J. Physiol. (London)*, Vol. 178, pp. 477–504, 1965.

[Bastos et al. 95] J. Bastos, M. Steyaert, R. Roovers, P. Kinget, W. Sansen, B. Graindourze, A. Pergoot & Er. Janssens, "Mismatch characterization of small size MOS transistors," *in Proc. IEEE Int. Conference on Microelectronic Test Structures*, pp. 271–276, March 1995.

[Bastos et al. 97] J. Bastos, M. Steyaert, A. Pergoot & W. Sansen, "Mismatch characterization of submicron MOS transistors," *Analog Integrated Circuits and Signal Processing*, Vol. 12, No. 2, pp. 95–106, February 1997.

[Beare and Bouzerdoum 96] R. Beare & A. Bouzerdoum, "A simple model of the SUSTAINED unit in the insect lamina," *Proc. Australian New Zealand Conference on Intelligent Information Systems*, pp. 109–112, 18-20 Nov., Adelaide, Australia 1996.

[Bernard et al. 93] T.M. Bernard, P.E. Nguyen, F.J. Devos & B.Y. Zavidovique, "A programmable VLSI retina for rough vision," *Machine Vision and Applications*, Vol. 7, No. 1, pp. 4–11, Winter 1993.

[Boahen and Andreou 92] K.A. Boahen & A.G. Andreou, "A contrast sensitive silicon retina with reciprocal synapses," *Advances in Neural Information Processing 4*, Vol. 4, pp. 762–772, 1992.

[Boahen 96] K.A. Boahen, "Retinomorphic vision systems," *Proc. 5th Int. Conf. Microelectronics for Neural Networks and Fuzzy Systems MicroNeuro'96*, pp. 2–14, 1996.

[Bohrer et al. 90] S. Bohrer, H.H. Bulthoff & H.A. Mallot, "Motion detection by correlation and voting," *Parallel Processing in Neural Systems and Computers*, pp. 471–474, 1990.

[Bossomaier and Snyder 85] T.R.S. Bossomaier & A.W. Snyder, "Irregularity and aliasing: solution?," *Vision Research*, Vol. 25, No. 1, pp. 145–147, 1985.

[Bouzerdoum and Pinter 89] A. Bouzerdoum & R.B. Pinter, "Image motion processing in biological and computer vision systems," *Proc. SPIE, Visual Communications and Image Processing IV*, Vol. 1199, pp. 1229–1240, 1989.

[Bouzerdoum and Pinter 92] A. Bouzerdoum & R.B. Pinter, "Nonlinear lateral inhibition applied to motion detection on the fly visual system," In R.B. Pinter & B. Nabet, editeurs, Nonlinear Vision, pp. 423–450, CRC press, 1992.

[Bouzerdoum et al. 94] A. Bouzerdoum, A. Moini, A. Yakovleff, X.T. Nguyen, R.E. Bogner & K. Eshraghian, "A smart visual micro-sensor," *Proc. Int. IEEE Conf. on Systems, Man, and Cybernetics*, pp. 276–279, 1994.

[Buxton and Buxton 84] B.F Buxton & H. Buxton, "Computation of optic flow from the motion of edge features in image sequences," *Image and Vision Computing*, Vol. 2, No. 2, pp. 59–75, May 1984.

[Chen and Ginosar 95] S. Chen & R. Ginosar, "Adaptive sensitivity CCD image sensor," *Proc. SPIE, Charge Coupled Devices and Solid State Optical Sensors V*, Vol. 2415, pp. 303–309, 1995.

[Chua and Yang 88a] L.O. Chua & L. Yang, "Cellula neural networks: applications," *IEEE Trans. Circuits & Sys.*, Vol. 35, No. 10, pp. 1273–1290, October 1988.

[Chua and Yang 88b] L.O. Chua & L. Yang, "Cellula neural networks: theory," *IEEE Trans. Circuits & Sys.*, Vol. 35, No. 10, pp. 1257–1272, October 1988.

[Corghi et al. 97] A. Corghi, R. Leonardi, J.M. Corridoni & A.Delbimbo, "Determining the optical flow using wavelet coefficients," *Proc. SPIE, Visual Communications and Image Processing '97*, Vol. 3024, pp. 315–327, 12-14 Feb. 1997.

[Courellis and Marmarelis 90] S.H. Courellis & V.Z. Marmarelis, "An artificial neural network for motion detection and speed estimation," *Proc. Int. Joint Conf. Neural Networks*, pp. 407–421, 1990.

[Crowley and Vardaman 94] R.T. Crowley & E.J. Vardaman, "Packaging alternatives with 3D technology," *Electronic Packaging and Production*, Vol. 4, No. 5, pp. 76–78, May 1994.

[Davari et al. 95] B. Davari, R.H. Dennard & G.G. Shahidi, "CMOS scaling for high performance and low power–the next ten years," *IEEE Proceedings*, Vol. 83, No. 4, pp. 595–606, April 1995.

[Delbrück 93] Tobias Delbrück. *Investigations of analog VLSI visual transduction and motion processing*. PhD thesis, California Institute of Technology, Pasadena, California, 1993.

[Dominguez-Castro et al. 97] R. Dominguez-Castro, S. Espejo, A. Rodriguez-Vazquez & R. Carmona, "A 0.8μm CMOS tw-dimensional programmable mixed-signal focal-plane array processor with on-chip binary imaging and instructions storage," *IEEE Journal of Solid State Circuits*, Vol. 32, No. 7, July 1997.

[Dron McIlrath 96] L. Dron McIlrath, "A CCD/CMOS focal-plane array edge detection processor implementing the multiscale veto algorithm," *IEEE Journal of Solid State Circuits*, Vol. 31, No. 9, pp. 1239–1247, September 1996.

[Dron 93] L. Dron, "Multiscale veto model: a two-stage analog network for edge detection and image reconstruction," *International Journal of Computer Vision*, Vol. 11, No. 1, pp. 45–61, August 1993.

[Dubois 85] R. Dubois, "The sampling and reconstruction of time-varying imagery with application in video systems," *IEEE Proceedings*, Vol. 73, No. 4, pp. 502–522, 1985.

[Eberly et al. 90] D. Eberly, D. Wenzel & H. Longbotham, "Hexagonal tessellations in image algebra," *Proc. SPIE, Image Algebra and Morphological Image Processing*, Vol. 1350, pp. 25–30, 1990.

[Espejo et al. 94] S. Espejo, A. Rodriguez-Vazquez, R. Dominguez-Castro & J.L.E. Sanchez-Sinencio, "Smart-pixel cellular neural networks current-mode CMOS technology," *IEEE Journal of Solid State Circuits*, Vol. 29, No. 8, pp. 895–905, August 1994.

[Fatt 78] I. Fatt. Physiology of the eye. Butterworth Publishers Inc., Boston, 1978.

[Faugeras 93] Olivier Faugeras. Three-dimensional computer vision : a geometric viewpoint. The MIT Press, Cambridge, Massachusetts, 1993.

[Forchheimer and Odmark 83] R. Forchheimer & A Odmark, "A single chip linear array processor," *Proc. SPIE, Applications of Digital Image Processing*, Vol. 397, pp. 425–430, 1983.

[Forti and Wright 94] F. Forti & M.E. Wright, "Measurement of MOS current mismatch in the weak inversion region," *IEEE Journal of Solid State Circuits*, Vol. 29, No. 2, pp. 138–142, 1994.

[Fowler et al. 94] B. Fowler, A. El Gamal & D.X.D. Yang, "A CMOS area image sensor with pixel-level A/D conversion," *Proc. IEEE Int. Solid State Circuits Conf.*, pp. 226–227, 1994.

[Fowler 95] Boyd Fowler. *CMOS Area Image Sensors with Pixel Level A/D Conversion*. PhD thesis, Stanford University, 1995.

[Franceschini et al. 89] N. Franceschini, A. Reihle & A. Le Nestour, "Directionally selective motion detection by insect neurons," In D.G. Stavenga & R.C. Hardie, editeurs, Facts of Vision, pp. 360–389, Springer-Verlag, Berlin, 1989.

[Fujita et al. 93] M. Fujita*et al.*, "Development and fabrication of digital neural network WSIS," *Journal of the Institute of Electronics, Information and Communication Engineers, Japan*, Vol. E76-C, No. 7, pp. 1182–1190, July 1993.

[Fukushima et al. 70] K. Fukushima, Y. Yamaguchi, M. Yasuda & S. Nagata, "An electronic model of the retina," *IEEE Proceedings*, Vol. 58, No. 12, pp. 1950–1951, December 1970.

[Furth and Andreou 95] P.M. Furth & A.G. Andreou, "Linearized differential transconductors in subthreshold CMOS," *Electronic Letters*, Vol. 31, No. 7, pp. 545–547, March 1995.

[Gabor 45] D. Gabor, "Theory of communications," *Journal of IEEE*, Vol. 93, pp. 429–456, 1945.

[Goles 90] E. Goles. Neural and automata networks : dynamical behavior and applications. Kluwer Academic, Dordrecht ; Boston, 1990.

[Gupta and Kanal 97] N. Gupta & L. Kanal, "Gradient based image motion estimation without computing gradients," *International Journal of Computer Vision*, Vol. 22, No. 1, pp. 81–101, Feb.-March 1997.

[Hahm and Titlebaum 97] M.D. Hahm & E.G. Friedman E.L. Titlebaum, "A comparison of analog and digital circuit implementations of low power matched filters for use in portable wireless communication terminals," *IEEE Trans. Circuits & Sys. II: Analog and Digital Signal Processing*, Vol. 44, No. 6, pp. 498–506, June 1997.

[Hamamoto et al. 96] T. Hamamoto, K. Aizawa & M. Hatori, "Motion adaptive image sensor for enhancement and wide dynamic range," *Proc. SPIE, Advanced Focal Plane Arrays and Electronic Cameras*, Vol. 2950, pp. 137–145, 1996.

[Hamamoto et al. 97] T. Hamamoto, K. Aizawa & M. Hatori, "Focal plane compression and enhancement sensors," *Proc. IEEE Int. Symposium on Circuits and Systems*, pp. 1912–1915, 1997.

[Hammerstorm 90] D. Hammerstorm, "A VLSI architecture for high-performance, low-cost, on-chip learning," *Proc. Int. Joint Conf. Neural Networks*, pp. 537–544, 1990.

[Haralick 93] Robert M. Haralick. Computer and robot vision, volume 1. Addison-Wesley Pub. Co., Reading, Massachusetts, 1993.

[Harris et al. 89] J. Harris, C. Koch, J. Luo & J. Wyatt, "Resistive fuses: Analog hardware for detecting discontinuities in early vision," In C. Mead & M. Isamil, editeurs, Analog VLSI Implementation of Neural Systems, pp. 27–56, Kluwer Academics, 1989.

[Harris et al. 90] J.G. Harris, C. Koch & J. Luo, "A two-dimensional analog VLSI circuit for detecting discontinuities in early vision," *Science*, Vol. 248, pp. 1209–1211, June 1990.

[Harris 91] J.G. Harris, "Discarding outliers using a nonlinear resistive network," *Proc. Int. Joint Conf. Neural Networks*, pp. 239–246, 1991.

[Hassenstein and Reichardt 56] B. Hassenstein & W. Reichardt, "Structure of a mechanism of perception of optical movement," *Proc. 1st International Congress on Cybernetics*, pp. 797–801, 1956.

[Hildreth 85] E.C. Hildreth. The measurement of visual motion. The MIT Press, Cambridge, Massachusetts, 1985.

[Hirotsu and Brooke 93] K. Hirotsu & M.A. Brooke, "Analog neural network chip with random weight change learning algorithm," *Proc. Int. Joint Conf. Neural Networks*, Vol. 3, pp. 3031–3034, 1993.

[Hochet et al. 90] B. Hochet, V. Peiris, G. Corbaz & M. Declerq, "Implementation of a neuron dedicated to kohonen maps with learning capabilities," *Proc. Custom Integrated Circuits Conf.*, pp. 26.1.1–26.1.4, 1990.

[Horn and Schunck 81] B.K.P. Horn & B.G. Schunck, "Determining optical flow," *Artificial Intelligence*, Vol. 17, pp. 185–203, 1981.

[Horn 86] Berthold Klaus Paul Horn. Robot vision. The MIT Press, Cambridge, Massachusetts, 1986.

[Horridge 75] G.A. Horridge. The compound eye and vision of insects. Clarendon Press, Oxford, 1975.

[Horridge 90] G.A. Horridge, "A template theory to relate visual processing to digital circuitry," *Proc. R. Soc. Lond.*, Vol. B-239, pp. 17–33, 1990.

[Hosticka 85] B.J Hosticka, "Performance comparison of analog and digital circuits," *IEEE Proceedings*, Vol. 73, No. 1, pp. 25–29, January 1985.

[Ibrahim and Zaghloul 90] F. Ibrahim & M.E. Zaghloul, "Design of modifiable-weight synapse CMOS analog cell," *Proc. IEEE Int. Symposium on Circuits and Systems*, Vol. 4, pp. 2978–2981, 1990.

[Inigo et al. 90] R.M. Inigo, A. Bonde & B. Holcombe, "Self adjusting weights for hardware neural networks," *Electronic Letters*, Vol. 26, No. 19, pp. 1630–1632, September 1990.

[Ismail and Fiez 94] M. Ismail & T. Fiez. Analog VLSI signal and information processing. McGraw-Hill, 1994.

[Kinget and Steyaert 96] P. Kinget & M. Steyaert, "Impact of transistor mismatch on the speed-accuracy-power trade-off of analog CMOS circuits," *Proc. Custom Integrated Circuits Conf.*, pp. 333–336, 1996.

[Kioi et al. 92] K. Kioi, T. Shinozaki, S. Toyoyama, K. Shirakawa, K. Ohtake & S. Tsuchimoto, "Design and implementation of a 3D LSI image sensing processor," *IEEE Journal of Solid State Circuits*, Vol. 27, No. 8, pp. 1130–1140, August 1992.

[Kobayashi et al. 91] H. Kobayashi, L. White & A.A. Abidi, "An active resistor network for gaussian filtering of images," *IEEE Journal of Solid State Circuits*, Vol. 26, No. 5, pp. 738–748, May 1991.

[Kondo et al. 94] Y. Kondo*et al.*, "1.2GFLOPS neural network chip exhibiting fast convergence," *Proc. IEEE Int. Solid State Circuits Conf.*, pp. 218–219, 1994.

[Lakshmikumar et al. 86] K.R. Lakshmikumar, R.A. Hadaway & M.A. Copeland, "Characterisation and modeling of mismatch in MOS transistors for precision analog design," *IEEE Journal of Solid State Circuits*, Vol. 21, No. 6, pp. 1057–1066, December 1986.

[Laplante and Stoyenko 96] P.A. Laplante & A.D. Stoyenko. Real-time imaging, chapter 2, pp. 27–58. IEEE Press, 1996.

[Larcombe et al. 95] S.P. Larcombe, J.M. Stern, P.A. Ivey & L. Seed, "Utilizing a low cost 3D packaging technology for consumer applications," *IEEE Trans. Consumer Electronics*, Vol. 41, No. 4, pp. 1095–1102, December 1995.

[Lazzaro and Wawrzynek 95] J. Lazzaro & J. Wawrzynek, "A multi-sender asynchronous extension to the AER protocol," *Proc. Advanced Research in VLSI*, pp. 158–169, 1995.

[Lazzaro et al. 93] J.P. Lazzaro, J. Wawrzynek, M. Mahowald.and M. Sivilotti & D. Gillespie, "Silicon auditory processors as computer peripherals," *IEEE Trans. Neural Networks*, Vol. 4, No. 3, pp. 523–528, 1993.

[LeQuang and Zaccarin 97] D. LeQuang & A. Zaccarin, "Motion field and image intensity segmentation for object-oriented coding of video sequences," *Proc. SPIE, Visual Communications and Image Processing '97*, Vol. 3024, pp. 711–722, 12-14 Feb. 1997.

[Linares-Barranco et al. 93] B. Linares-Barranco*et al.*, "A CMOS analog adaptive BAM with on-chip learning and weight refreshing," *IEEE Trans. Neural Networks*, Vol. 4, No. 3, pp. 445–455, May 1993.

[Lumsdaine et al. 91] A. Lumsdaine, J.L. Wyatt & I.M. Elfadel, "Nonlinear analog networks for image smoothing and segmentation," *Journal of VLSI Signal Processing*, Vol. 3, pp. 53–68, 1991.

[Lyon 81] R.F. Lyon, "The optical mouse, and an architectural methodology for smart digital sensors," *In Proc. CMU Conference on VLSI Structures and Computations*. Computer Science Press, October 1981.

[Macq et al. 93] D. Macq, M. Verlcysen, P. Jespers & J.D. Legat, "Analog implementation of a kohonen map with on-chip learning," *IEEE Trans. Neural Networks*, Vol. 4, No. 3, pp. 456–461, May 1993.

[Makie-Fukuda et al. 95] K. Makie-Fukuda, S. Maeda, T. Tsukada & T. Matsuura, "Substrate noise reduction using active guard band filters in mixed-signal integrated circuits," *Symposium on VLSI Circuits, Kyoto, 8-10 June*, pp. 33–34, 1995.

[Marr and Hildreth 80] D. Marr & E. Hildreth, "Theory of edge detection," *Proc. of Royal Society of London*, Vol. B 200, pp. 269–274, 1980.

[Masui 92] S. Masui, "Simulation of substrate coupling in mixed-signal MOS circuits," *Symposium on VLSI Circuits, Seattle, 4-6 June*, pp. 42–43, 1992.

[Matsumoto et al. 93] T. Matsumoto, H. Kobayashi & T. Yagi, "Vision chip. II. analog image-processing neuro chip," *Journal of the Institute of Electronics, Information and Communication Engineers, Japan*, Vol. 76, No. 8, pp. 851–858, Aug. 1993.

[Mead and Ismail 89] C. Mead & M. Ismail. Analog VLSI implementation of neural systems. Kluwer Academic Publishers, Boston, 1989. Proceedings of a workshop on Analog Integrated Neural Systems.

[Mead and Mahowald 88] C. Mead & M.A. Mahowald, "A silicon model of early visual processing," *Neural Networks*, Vol. 1, pp. 91–97, 1988.

[Mead 89] C. Mead. Analog VLSI and neural systems. Addison-Wesley, Reading, Massachusetts, 1989.

[Memin and Perez 96] E. Memin & P. Perez, "Multiresolution markov random field and multigrid algorithm for discontinuity preserving estimation of the optical flow," *Proc. SPIE, Neural, Morphological, and Stochastic Methods in Image and Signal Processing*, Vol. 2568, pp. 30–41, 1996.

[Moini and Bouzerdoum 97] A. Moini & A. Bouzerdoum, "A biologically motivated imager and motion detector with pixel level image processing," *Australian Microelectronics Conference*, pp. 180–185, 1997.

[Moini et al. 93] A. Moini, A. Bouzerdoum, A. Yakovleff, D. Abbott, O. Kim, K. Eshraghian & R.E. Bogner, "An analog implementation of early visual processing in insects," *Proc. 1993 Int. Symposium on VLSI Technology, Systems, and Applications*, pp. 283–287, May 1993.

[Moini et al. 95a] A. Moini, A. Blanksby, R. Beare, A. Bouzerdoum & K. Eshraghian, "Multiplicative noise cancellation (MNC) in analog VLSI vision sensors," *Electronics Technology Directions for the Year 2000 (ETD2000), Adelaide, Australia*, pp. 253–257, 1995.

[Moini et al. 95b] A. Moini, A. Bouzerdoum, K. Eshraghian, A. Yakovleff & X.T. Nguyen, "The architecture of an insect vision based VLSI motion detection chip," *Australian Microelectronics Conference*, pp. 68–73, 1995.

[Moini et al. 96] A. Moini, A. Bouzerdoum, A. Yakovleff & K. Eshraghian, "A two dimensional motion detector based on the insect vision," *Proc. SPIE, Advanced Focal Plane Arrays and Electronic Cameras, Berlin*, pp. 146–157, 10-11 Oct. 1996.

[Moini et al. 97a] A. Moini, A. Bouzerdoum & K. Eshraghian, "A current mode implementation of shunting inhibition," *Proc. IEEE Int. Symposium on Circuits and Systems*, pp. 557–560, 1997.

[Moini et al. 97b] A. Moini, A. Bouzerdoum, K. Eshraghian, A. Yakovleff, X.T. Nguyen, A. Blanksby, R. Beare, D. Abbott & R.E. Bogner, "An insect vision-based motion detection chip," *IEEE J. Solid State Circuits*, Vol. 32, No. 2, pp. 279–284, February 1997.

[Moini 94] Alireza Moini. Design of a VLSI motion detector based upon the insect visual system. Master's thesis, The University of Adelaide, 1994.

[Moon et al. 92] G. Moon, M.E. Zaghloul & R.W. Newcomb, "VLSI implementation of synaptic weighting and summing in pulse coded neural-type cells," *IEEE Trans. Neural Networks*, Vol. 3, No. 3, pp. 394–403, May 1992.

[Morishita et al. 90] T. Morishita, Y. Tamura & T. Otsuki, "A BiCMOS analog neural network with dynamically updated weights," *Proc. IEEE Int. Solid State Circuits Conf.*, pp. 142–143, 1990.

[Morishita et al. 93] T. Morishita*et al.*, "Digital neural network coprocessor with a dynamically reconfigurable pipeline architecture," *Journal of the Institute of Electronics, Information and Communication Engineers, Japan*, Vol. E76-C, No. 7, pp. 1191–1196, July 1993.

[Murray and Tarassenko 94] A. Murray & L. Tarassenko. Analogue neural vlsi, a pulse stream approach. Chapman & Hall, London ; New York,, 1994.

[Musmann et al. 85] H.G. Musmann, P. Pirsch & H.-J. Grallert, "Advances in picture coding," *IEEE Proceedings*, Vol. 73, No. 4, pp. 523–548, April 1985.

[Nabet and Pinter 91] B. Nabet & R.B. Pinter. Sensory neural networks: Lateral inhibition. CRC Press Inc., 1991.

[Nagel 87] H.-H. Nagel, "On the estimation of optic flow: Relations between different approaches and some new results," *Artificial Intelligence*, Vol. 33, No. 3, pp. 299–324, November 1987.

[Nagel 90] H.-H. Nagel, "Extending the 'oriented smoothness constraint' into the temporal domain and the estimation of derivatives of optical flow," *In Proc. 1st European Conf. on Computer Vision*, pp. 139–148, 1990.

[Nakahira et al. 93] H. Nakahira*et al.*, "Digital neuroprocessor using quantizer neurons," *Symposium on VLSI Circuits*, pp. 35–36, 1993.

[Newcomb 81] R.W. Newcomb, "Neural-type microsystems-circuit status," *In Proc. IEEE Int. Symposium on Circuits and Systems*, volume 1, pp. 97–100, Chicago, Illinoise, USA, 27–29 April 1981.

[Nguyen et al. 93] X.T. Nguyen, K. Eshraghian, A. Moini, A. Bouzerdoum, A. Yakovleff, D. Abbott & R.E. Bogner, "An implementation of a smart visual micro-sensor based

upon insect vision," *12th Australian Microelectronics Conference*, pp. 129–134, 1993.

[Nguyen et al. 94] X.T. Nguyen, A. Bouzerdoum, A. Yakovleff, A. Moini, R.E. Bogner & K. Eshraghian, "A VLSI robotic micro-sensor: Range-finder from self motion," *Proc. Int. Conf. on Machine Vision in Practice, Toowoomba, Australia*, pp. 78–83, September 1994.

[Nguyen 96] Xuan Thong Nguyen. *Smart VLSI Micro-Sensors for Velocity Estimation Inspired by Insect Vision*. PhD thesis, Department of Electrical & Electronics Engineering, The University of Adelaide, SA 5005, Australia, 1996.

[Ni et al. 96] Y. Ni, B. Arion & F. Devos, "An analog retina based stereo vision system for car navigation," *Proc. SPIE, Advanced Focal Plane Arrays and Electronic Cameras, Berlin*, pp. 274–281, 10-11 Oct. 1996.

[Nilson et al. 94] C.D. Nilson, R.B. Darling & R.B. Pinter, "Shunting neural network photodetector arrays in analog CMOS," *IEEE Journal of Solid State Circuits*, Vol. 29, No. 10, pp. 1291–1296, October 1994.

[Nitta et al. 92a] Y. Nitta*et al.*, "Optical neurochip with learning capability," *IEEE Trans. Photonics Technology Letters*, Vol. 4, No. 3, pp. 247–249, March 1992.

[Nitta et al. 92b] Y. Nitta, J. Ohta, M. Takahashi, S. Tai & K. Kyuma, "Optical neurochip with learning capability," *IEEE Photonics Technology Letters*, Vol. 4, No. 3, pp. 247–249, Mar 1992.

[Nitta et al. 93] Y. Nitta, J. Ohta, S. Tai & K. Kyuma, "Monolithic integration of optical neurochip with variable sensitivity photodetector," *IEEE Photonics Technology Letters*, Vol. 5, No. 1, pp. 67–69, January 1993.

[Nitta et al. 95] Y. Nitta, J. Ohta, S. Tai & K. Kyuma, "Optical neurochip for image processing," *Electronics and Communications in Japan, Part 2 (Electronics)*, Vol. 78, No. 1, pp. 10–20, Jan. 1995.

[Ong and Spann 96] E.P. Ong & M. Spann, "Robust multiresolution computation of optical flow," *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Vol. 4, pp. 1938–1941, 1996.

[Oteki et al. 93] S. Oteki*et al.*, "Digital neural network VLSI with on-chip learning using stochastic pulse encoding," *Proc. Int. Joint Conf. Neural Networks*, Vol. 3, pp. 3039–3045, 1993.

[Pardo et al. 97] F. Pardo, B. Dierickx & D. Scheffer, "CMOS foveated image sensor: signal scaling and small geometry effects," *IEEE Trans. Electron Devices*, Vol. 44, No. 10, pp. 1731–1737, October 1997.

[Pastore and Premoli 97] S. Pastore & A. Premoli, "Finding all dc solutions of nonlinear resistive circuits by exploring both polyhedral and rectangular circuits," *IEE Proceedings-Circuits, Devices and Systems*, Vol. 144, No. 1, pp. 17–21, feb 1997.

[Pavasovic et al. 94] A. Pavasovic, A.G. Andreou & C.R. Westgate, "Characterization of subthreshold MOS mismatch in transistors for VLSI systems," *Analog Integrated Circuits and Signal Processing*, Vol. 6, No. 1, pp. 75–85, July 1994.

[Pelgrom et al. 89] M.J.M. Pelgrom, A.C.J. Duinmaijer & A.P.G. Welbers, "Matching properties of MOS transistors," *IEEE Journal of Solid State Circuits*, Vol. 24, No. 5, pp. 1433–1439, October 1989.

[Pinter 84] R.B. Pinter, "Adaptation of receptive field spatial organization via multiplicative lateral inhibition," *IEEE Int. Conf. on Systems, Man, and Cybernetics*, pp. 328–331, 1984.

[Raffo 95] L. Raffo, "Resistive network implementing maps of Gabor functions of any phase," *Electronic Letters*, Vol. 31, No. 22, pp. 1913–1914, 1995.

[Raffo 96] L. Raffo, "Analysis and synthesis of resistive networks for distributed visual elaborations," *Electronic Letters*, Vol. 32, No. 8, pp. 743–744, 1996.

[Ratliff and Hartline 74] F. Ratliff & H.K. Hartline. Studies on excitation and inhibition in the retina. Chapman and Hall, London, 1974.

[Regan 91] D. Regan, "A brief review of some of the simuli and analysis methods used in spatiotemporal vision research," In D. Regan, editeur, Vision and Visual Dysfunction, Volume 10: Spatial Vision, chapter 1, pp. 1–42, CRC Press, Inc., 1991.

[Reichardt and Egelhaaf 88] W. Reichardt & M. Egelhaaf, "Properties of individual movement detectors as derived from behavioural experiments on the visual system of the fly," *Biological Cybernetics*, Vol. 58, No. 5, pp. 287–294, 1988.

[Roska and Chua 93] T. Roska & L.O. Chua, "The CNN universal machine: an analogic array computer," *IEEE Trans. Circuits & Sys. II: Analog and Digital Signal Processing*, Vol. 40, No. 3, pp. 163–173, March 1993.

[Runge et al. 68] R.G. Runge, M. Uemura & S.S. Viglione, "Electronic synthesis of the neural networks in the pigeon retina," *In Cybernetic problems in bionics*, pp. 791–800, Dayton, Ohio, USA, 3-5 May 1968.

[Salam and Choi 90] F.M.A. Salam & M.R. Choi, "An all-MOS analog feedforward neural circuit with learning," *Proc. IEEE Int. Symposium on Circuits and Systems*, Vol. 4, pp. 2508–2511, 1990.

[Sarpeshkar et al. 96a] R. Sarpeshkar, J. Kramer, G. Indiveri & C. Koch, "Analog VLSI architectures for motion detection: From fundamental limits to system applications," *IEEE Proceedings*, Vol. 84, No. 7, pp. 969–987, July 1996.

[Sarpeshkar et al. 96b] R. Sarpeshkar, R.F. Lyon & C.A. Mead, "An analog VLSI cochlea with new transconductance amplifiers and nonlinear gain control," *Proc. IEEE Int. Symposium on Circuits and Systems*, Vol. 3, pp. 292–295, 1996.

[Sarpeshkar 97] Rahul Sarpeshkar. *Efficient precise computation with noisy components: extrapolating from an electronic cochlea to the brain.* PhD thesis, California Institute of Technology, Pasadena, California, 1997.

[Sato et al. 93] Y. Sato*et al.*, "Development of a high-performance, general purpose neurocomputer composed of 512 digital neurons," *Proc. Int. Joint Conf. Neural Networks*, pp. 1967–1970, 1993.

[Schmid and Bulthoff 88] A. Schmid & H. Bulthoff, "Using neuropharmacology to distinguish between excitatory and inhibitory movement detection mechanisms in the fly *calliphora erythrocephala*," *Biological Cybernetics*, Vol. 59, No. 2, pp. 71–80, 1988.

[Seevinck 88] Evert Seevinck. Analysis and synthesis of translinear integrated circuits. Elsevier, Amsterdam, New York, 1988.

[Seevinck 90] E. Seevinck, "Companding current-mode integrator, a new circuit principle for continuous-time monolithic filters," *Electronic Letters*, Vol. 26, No. 24, pp. 2046–2047, 1990.

[Shi and Chua 92] B.E. Shi & L.O. Chua, "Resistive grid image filtering: input/output analysis via the CNN framework," *IEEE Trans. Circuits & Sys. I: Fundamental Theory and Applications*, Vol. 39, No. 7, pp. 531–548, July 1992.

[Shyu et al. 84] J.-B. Shyu, G.C. Temes & F. Krummenacher, "Random error effects in matched MOS capacitors and current sources," *IEEE Journal of Solid State Circuits*, Vol. SC-19, No. 6, pp. 948–956, December 1984.

[Smith and Brady 97] S.M. Smith & J.M. Brady, "SUSAN-a new approach to low level image processing," *International Journal of Computer Vision*, Vol. 23, No. 1, pp. 45–78, 1997.

[Srinivasan et al. 82] M.V. Srinivasan, S.B. Laughlin & A. Dubs, "Predictive coding: a fresh view of inhibition in the retina," *Proc. R. Soc. Lond.*, Vol. B 216, pp. 427–459, 1982.

[Stern et al. 96] J.M. Stern, S.P. Larcombe, P.A. Ivey, L. Seed, A.J. Shelley & N.J. Goodenough, "Design and evaluation of an epoxy three-dimensional multichip module," *IEEE Trans. Components, Packaging and Manufacturing Technology, Part B: Advanced Packaging*, Vol. 19, No. 1, pp. 188–194, February 1996.

[Steyaert et al. 94] M. Steyaert, J. Bastos, R. Roovers, P. Kinget, W. Sansen, B. Graindourze, A. Pergoot & Er. Janssens, "Threshold voltage mismatch in short-channel MOS transistors," *Electronic Letters*, Vol. 18, pp. 1546–1548, September 1994.

[Su et al. 93] D.K. Su, M.J. Loinaz, S. Masui & B.A. Wooley, "Experimental results and modeling techniques for substrate noise in mixed-signal integrated circuits," *IEEE Journal of Solid State Circuits*, Vol. 28, No. 4, pp. 420–430, April 1993.

[Tam et al. 90] S.M. Tam, B. Gupta, H.A. Castro & M. Holler, "Learning on an analog VLSI neural network chip," *Proc. IEEE Int. Conf. Systems, Man and Cybernetics*, pp. 701–703, 1990.

[Tanner and Mead 84] J. Tanner & C. Mead, "A correlating optical motion detector," *In MIT Advanced Research in VLSI*, pp. 57–64, 1984.

[Tawel 93] R. Tawel, "Learning in analog neural network hardware," *Computers & Electrical Engineering*, Vol. 19, No. 6, pp. 453–467, November 1993.

[Terzopolous 84] Demetri Terzopolous. *Multiresolution computation of visible-surface representations*. PhD thesis, Dept. of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1984.

[Tomazou et al. 90] C. Tomazou, F.J. Lindgey & D.G. Haigh. Analog IC design: the current-mode approach. Peter Peregrinus Ltd., Londong, United Kingdom, 1990.

[Torelli et al. 96] G. Torelli, L. Gonzo, M. Gottardi, F. Maloberti, A. Sartori & A. Simoni, "Analog-to-digital conversion architectures for intelligent optical sensor arrays," *Proc. SPIE, Advanced Focal Plane Arrays and Electronic Cameras, Berlin*, Vol. 2950, pp. 254–264, 1996.

[Tsividis 97] Y. Tsividis, "Externally linear, time-invariant systems and their application to companding signal processors," *IEEE Trans. Circuits & Sys. II: Analog and Digital Signal Processing*, Vol. 44, No. 2, pp. 65–85, February 1997.

[Umminger and Sodini 92] C.B. Umminger & C.G. Sodini, "Switched capacitor networks for focal plane image processing systems," *IEEE Trans. Circuits & Systems for Video*, Vol. 2, No. 4, pp. 392–400, December 1992.

[van der Spiegel et al. 89] Jan van der Spiegel, G. Kreider, C. Claeys, I. Debusschere, G. Sandini, P. Dario, F. Fantini, P. Belluti & G. Soncini, "A foveated retina-like sensor using CCD technology," In C. Mead & M. Ismail, editeurs, Analog VLSI implementation of neural systems, chapter 8, pp. 189–212, Kluwer Academic Publishers, Boston, 1989. Proceedings of a workshop on Analog Integrated Neural Systems.

[Venier et al. 96] P. Venier, O. Landolt, P. Debergh & X. Arreguit, "Analog CMOS photosensitive array for solar illumination monitoring," *Proc. IEEE Int. Solid State Circuits Conf.*, pp. 96–97, 1996.

[Verghese et al. 96] N.K. Verghese, D.J. Allstot & M.A. Wolfe, "Verification techniques for substrate coupling and their application to mixed-signal ic design," *IEEE Journal of Solid State Circuits*, Vol. 31, No. 3, pp. 345–365, March 1996.

[Vidal 83] J.J. Vidal, "Silicon brains: Whither neuromimetic computer architectures," *In Proc. IEEE International Conference on Computer Design: VLSI in Computers (ICCD '83)*, pp. 17–20, Port Chester, NY, USA, 31 Oct.– 3 Nov. 1983.

[Vittoz et al. 91] E.A. Vittoz, O. Oguey, M.A. Maher, O. Nys, E. Dijkstra & M. Chevroulet, "Analog storage of adjustable synaptic weights," In U. Ramacher & U. Ruckert, editeurs, VLSI Design of Neural Networks, pp. 47–63, Kluwer Academics Publishers, Boston, MA, 1991.

[Wang et al. 93] T. Wang, X. Zhuang, Z. Xing & X. Xiao, "Neuron-weighted learning algorithm and its hardware implementation in associative memories," *IEEE Trans. Computers*, Vol. 42, No. 5, pp. 636–640, May 1993.

[Watson and Solomon 97] A.B. Watson & J.A. Solomon, "Model of visual contrast gain control and pattern masking," *Journal of the Optical Society of America A (Optics, Image Science and Vision)*, Vol. 14, No. 9, pp. 2379–2391, September 1997.

[Watson 91] A.B. Watson, "Neural contrast sensitivity," In M.S. Landy & J.A. Movshon, editeurs, Computational Models of Visual Processing, chapter 7, pp. 95–107, The MIT Press, Cambridge, Massachusetts, 1991.

[Wodnicki et al. 97] R. Wodnicki, G.W. Roberts & M.D. Levine, "A log-polar image sensor fabricated in a standard 1.2-μ m ASIC CMOS process," *IEEE Journal of Solid State Circuits*, Vol. 32, No. 8, pp. 1274–1277, August 1997.

[Yakovleff et al. 93] A. Yakovleff, A. Moini, A. Bouzerdoum, X.T. Nguyen, R.E. Bogner & K. Eshraghian, "A micro-sensor based on insect vision," *Workshop on Computer Architecture for Machine Perception*, pp. 137–146, December 1993.

[Yakovleff 96] Andre Yakovleff. *Architectural Studies For Vision Processing*. PhD thesis, Department of Electrical & Electronics Engineering, The University of Adelaide, SA 5005, Australia, 1996.

[Yamamura and Mishina 96] K. Yamamura & M. Mishina, "An algorithm for finding all solutions of piecewise-linear resistive circuits," *International Journal of Circuit Theory and Applications*, Vol. 24, No. 2, pp. 223–231, 1996.

[Yang et al. 96] D.X.D. Yang, A. El Gamal & B. Fowler, "A 128×128 pixel CMOS area image sensor with multiplexed pixel level A/D conversion," *Proc. Custom Integrated Circuits Conf.*, pp. 303–306, 1996.

[Yasunaga et al. 93] M. Yasunaga*et al.*, "A self-learning digital neural network using wafer-scale LSI," *IEEE Journal of Solid State Circuits*, Vol. 28, No. 2, pp. 106–14, Feb. 1993.

[Yellott Jr. 83] J.I. Yellott Jr., "Spectral consequences of photoreceptor sampling in the rhesus retina," *Science*, Vol. 221, No. 4608, pp. 382–385, 22 July 1983.

[Yellott Jr. 84] J.I. Yellott Jr., "Image sampling properties of photoreceptors: a reply to Miller and Bernard," *Vision Research*, Vol. 24, No. 3, pp. 281–282, 1984.

[Yu et al. 92] P.C. Yu, S.J. Decker, H.S. Lee & C.G. Sodini, "CMOS resistive fuses for image smoothing and segmentation," *IEEE Journal of Solid State Circuits*, Vol. 27, No. 4, pp. 545–553, April 1992.