

BIOMIMETIC VISUAL NAVIGATION ARCHITECTURES FOR
AUTONOMOUS INTELLIGENT SYSTEMS

by
Vivek Pant

Copyright © Vivek Pant, 2007

A Dissertation Submitted to the Faculty of the
ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT
In Partial Fulfillment of the Requirements
For the Degree of
DOCTOR OF PHILOSOPHY
In the Graduate College
THE UNIVERSITY OF ARIZONA

2007

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by **Vivek Pant**

entitled “**Biomimetic Visual Navigation Architectures for Autonomous Intelligent Systems**”

and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of **Doctor of Philosophy**

_____ Date: **November 26, 2007**
Charles M. Higgins, Ph.D.

_____ Date: **November 26, 2007**
Hal S. Tharp, Ph.D.

_____ Date: **November 26, 2007**
Janet M. Wang, Ph.D.

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

_____ Date: **November 26, 2007**
Dissertation Director: **Charles M. Higgins, Ph.D.**

STATEMENT BY AUTHOR

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

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

SIGNED: _____

APPROVAL BY DISSERTATION DIRECTOR

This dissertation has been approved on the date shown below:

Dr. Charles M. Higgins
Department of Electrical and Computer
Engineering

Date

ACKNOWLEDGEMENTS

I am greatly thankful to my parents and my sisters for their love and support without which none of this would have been possible. I am especially thankful to Mansi, my wife, for being unconditionally patient and loving during all the ups and downs of my graduate life.

I am grateful to my advisor Chuck Higgins for his guidance and encouragement throughout this work. I am also grateful to Prof. Hal S. Tharp and Prof. Janet Wang Roveda for serving on my dissertation defense committee.

My great thanks to Jon Dyhr for his friendship, encouragement, and help with improving this document. I would also like to thank Jon, Tim, and Ichi for all the uplifting discussions during coffee breaks.

TABLE OF CONTENTS

LIST OF FIGURES	7
ABSTRACT	9
CHAPTER 1. INTRODUCTION	10
1.1. The insect visual system	10
1.1.1. Optical-flow based models	12
1.1.2. Correlation-based models	12
1.1.3. Neuronally based EMD model	15
1.1.4. Contrast saturation	16
1.2. Visual navigation	16
1.3. Neuromorphic Engineering	17
1.4. Summary of the dissertation	18
CHAPTER 2. A COMPARATIVE STUDY OF BIOLOGICAL COLLISION AVOIDANCE MODELS	19
2.1. Collision avoidance models	19
2.2. The spatial and temporal integration based model	19
2.3. LGMD-DCMD neurons based models	20
2.3.1. Rind et al's LGMD based neural network	20
2.3.2. The η -function based mathematical model	23
2.4. Methods	25
2.4.1. Visual stimuli	25
2.4.2. Computational models	26
2.5. Results	30
2.5.1. Experiment 1	30
2.5.2. Experiment 2	31
2.6. Discussion	35
2.7. Summary	37
CHAPTER 3. THE EFFECT OF TRACKING ON COLLISION AVOIDANCE PERFORMANCE	38
3.1. Analysis of a 1D object's approach	38
3.2. Motion under 2D perspective projection	41
3.3. Collision avoidance models - an analysis	42
3.3.1. The Rind model	42
3.3.2. The STI model	43
3.4. Tracking and collision avoidance	45
3.5. Methods	47
3.6. Simulation results	49
3.6.1. Rind model without tracking	49
3.6.2. STI model without tracking	49
3.6.3. Rind model with tracking	51
3.6.4. STI model with tracking	51
3.6.5. Effect of noisy tracking	53
3.7. Summary	53

CHAPTER 4. CAMERA-BASED IMPLEMENTATION OF COLLISION AVOIDANCE MODELS	55
4.1. System details	55
4.2. Results	57
4.2.1. The STI algorithm	57
4.2.2. The Rind algorithm	63
4.3. Summary	66
CHAPTER 5. SPEED ESTIMATION BY NON-DIRECTIONAL MOTION MODELS	68
5.1. The NDM model	68
5.1.1. Response characteristics	70
5.1.2. Flicker response	73
5.2. The NDS model	73
5.2.1. Response characteristics	73
5.2.2. Flicker response	77
5.3. Summary	77
CHAPTER 6. ANALOG VLSI IMPLEMENTATION OF SPEED ESTIMATION SENSORS	79
6.1. Circuit architecture	79
6.1.1. Adaptive photoreceptor	81
6.1.2. Low-pass filter	82
6.1.3. Current-mode differential voltage addition circuit	83
6.1.4. Absolute value circuit	83
6.1.5. Differential add-multiply circuit	84
6.1.6. Scanner circuit	86
6.1.7. Current sense amplifier	86
6.2. Chip layout	86
6.3. Characterization	87
6.3.1. Differential add-multiply circuit	90
6.3.2. NDM pixel characterization	92
6.3.3. NDS pixel characterization	95
6.4. Discussion	95
6.5. Non-directional speed based navigation	97
6.6. Summary	98
CHAPTER 7. FINAL COMMENTS AND FUTURE WORK	100
7.1. Summary of contributions	100
7.1.1. Collision avoidance models with tracking	100
7.1.2. Non-directional speed-estimation model	101
7.2. Conclusions	103
REFERENCES	104

LIST OF FIGURES

FIGURE 1.1.	Diagram of the Insect Visual System	11
FIGURE 1.2.	Optical Flow Field	13
FIGURE 1.3.	The Hassenstein-Reichardt Model	14
FIGURE 1.4.	Elaborated EMD Model	15
FIGURE 2.1.	Tammero and Dickinson's model	21
FIGURE 2.2.	Rind and Bramwell's neural network architecture	22
FIGURE 2.3.	The η -function based computational model	24
FIGURE 2.4.	Continuous-time Rind model	27
FIGURE 2.5.	Continuous-time Rind model with motion input	29
FIGURE 2.6.	Response of the continuous-time Rind model to an approaching object	32
FIGURE 2.7.	Response of the continuous-time Rind model with rectification to multiple car videos	33
FIGURE 2.8.	Response of the continuous-time Rind model with motion input to multiple car videos	34
FIGURE 2.9.	Response of the Spatio-temporal integration model to multiple car videos	35
FIGURE 3.1.	A 1D object's approach towards a fixed observer	39
FIGURE 3.2.	2D projection of a 3D object	41
FIGURE 3.3.	Geometrical interpretation of an underdetermined system of equations	43
FIGURE 3.4.	Situations which limit the collision avoidance performance of the Rind model	44
FIGURE 3.5.	Limitations of the Spatial and Temporal Integration model	46
FIGURE 3.6.	Active Observer with two degrees of freedom	47
FIGURE 3.7.	Mean peak response of the Rind model versus the distance from the observer	50
FIGURE 3.8.	Mean peak response of the STI model versus the distance from the observer	50
FIGURE 3.9.	Mean peak response versus the distance from the observer for Rind model with tracking	52
FIGURE 3.10.	Mean peak response versus the distance from the observer for STI model with tracking	52
FIGURE 3.11.	Mean peak response versus the distance from the sensor plane for noisy tracking	53
FIGURE 4.1.	Experimental setup of the collision detection experiment	56
FIGURE 4.2.	A snapshot of the GUI used to control the collision detection algorithms	58
FIGURE 4.3.	Different approach trajectories	59
FIGURE 4.4.	Average response of the STI algorithm with and without tracking	60
FIGURE 4.5.	Experimental data from the STI algorithm with and without tracking	62
FIGURE 4.6.	Average response of the Rind algorithm with and without tracking	64
FIGURE 4.7.	Experimental data from the Rind algorithm with and without tracking	65
FIGURE 5.1.	Non-directional multiplication (NDM) based computational models	69
FIGURE 5.2.	Response of the NDM models against stimulus speed	71
FIGURE 5.3.	Contour plots of the mean response of NDM models	71
FIGURE 5.4.	Comparison between $1/\omega_x$ and $\cos(\omega_x)$	72
FIGURE 5.5.	Non-directional summation (NDS) based computational models	74
FIGURE 5.6.	Amplitude response of the NDS models against stimulus speed	75
FIGURE 5.7.	Contour plots of the amplitude response of NDS models	76
FIGURE 5.8.	Comparison between $1/\omega_x$ and $(1 + 2 \cos(\omega_x))$	76

FIGURE 5.9.	Comparison of the response of the simplified non-directional models to flicker and motion	78
FIGURE 6.1.	Block diagrams of the chip architecture	80
FIGURE 6.2.	Adaptive Photoreceptor circuit	81
FIGURE 6.3.	A g_m -C low-pass filter circuit	82
FIGURE 6.4.	A current-mode differential voltage addition circuit	83
FIGURE 6.5.	A current-mode absolute value circuit	84
FIGURE 6.6.	A differential add-multiply circuit	85
FIGURE 6.7.	A current sense amplifier circuit	86
FIGURE 6.8.	NDM pixel layout	87
FIGURE 6.9.	Juxtaposed NDS and NDM pixels	88
FIGURE 6.10.	The ND chip	89
FIGURE 6.11.	Chip characterization setup	89
FIGURE 6.12.	Circuit board for testing the sensor	90
FIGURE 6.13.	Response of the add-multiply circuit to V_{cd} and V_{ad}	91
FIGURE 6.14.	Response of the add-multiply circuit versus the input voltage at V_{bd} is varied for various values of V_{ad}	92
FIGURE 6.15.	Output characteristics of the differential add-multiply circuit	93
FIGURE 6.16.	Response of NDM pixels to a sinusoidal stimulus	94
FIGURE 6.17.	Response of NDS pixels to a sinusoidal stimulus	96
FIGURE 6.18.	Diagram of a sensor mounted robot	98
FIGURE 7.1.	Short-channel effect	102

ABSTRACT

Intelligent systems with even the bare minimum of sophistication require extensive computational power and complex processing units. At the same time, small insects like flies are adept at visual navigation, target pursuit, motionless hovering flight, and obstacle avoidance. Thus, biology provides engineers with an unconventional approach to solve complicated engineering design problems. Computational models of the neuronal architecture of the insect brain can provide algorithms for the development of software and hardware to accomplish sophisticated visual navigation tasks. In this research, we investigate biologically-inspired collision avoidance models primarily based on visual motion. We first present a comparative analysis of two leading collision avoidance models hypothesized in the insect brain. The models are simulated and mathematically analyzed for collision and non-collision scenarios. Based on this analysis it is proposed that along with the motion information, an estimate of distance from the obstacle is also required to reliably avoid collisions. We present models with tracking capability as solutions to this problem and show that tracking indirectly computes a measure of distance. We present a camera-based implementation of the collision avoidance models with tracking. The camera-based system was tested for collision and non-collision scenarios to verify our simulation claims that tracking improves collision avoidance. Next, we present a direct approach to estimate the distance from an obstacle by utilizing non-directional speed. We describe two simplified non-directional speed estimation models: the non-directional multiplication (ND-M) sensor, and the non-directional summation (ND-S) sensor. We also analyze the mathematical basis of their speed sensitivity. An analog VLSI chip was designed and fabricated to implement these models in silicon. The chip was fabricated in a $0.18\ \mu\text{m}$ process and its characterization results are reported here. As future work, the tracking algorithm and the collision avoidance models may be implemented as a sensor chip and used for autonomous navigation by intelligent systems.

CHAPTER 1

INTRODUCTION

The already significant challenge of designing intelligent systems is often made harder for engineers due to a trade-off between computational power and size. In artificial systems, packaging of sensors and their respective processing elements in a small area often comes at the price of reduced functionality. However, biological systems balance the act of sensor performance and size regulation to near perfection. This dissertation aims to augment our understanding of a biological system and develop an engineering solution based on its principles. Specifically, we are interested in understanding the biological mechanisms behind visually mediated obstacle avoidance and collision detection with the goal of designing a self-navigating robotic system that can avoid obstacles in its path.

Robotic visual navigation is a complex task that requires data to be acquired from an image sensor and processed in real time. Conventional computer architectures may be used for autonomous navigation in many cases. However, they require fast analog-to-digital converters (ADCs) to communicate the data from a high-resolution sensor (such as a CCD array) to a serial digital processor. They further demand a fast algorithm to process this data in order to perform realtime decisions. In a system that is required to operate on a limited power budget, such as an autonomous robot, the cost of high-speed data transfer and processing is often a limitation. An elegant alternative approach to both sensory data acquisition and processing may be found in the insect visual system. The sensors and the processing neurons in the insect brain are arranged in close proximity and data from each sensor is processed in parallel. This system has survived and evolved over hundreds of millions of years and provides us with highly optimized neuronal algorithms that can be used to design architectures for autonomous navigation.

In the next few sections, we present a brief overview of the insect visual system, some of the schemes that have been used to model motion detection and visual navigation in insects, and an introduction to a branch of engineering that implements these models as VLSI chips and artificially intelligent systems.

1.1 The insect visual system

The visual system is one of the most complicated processing architectures in the brain. It is no surprise then that in most animals, including primates, more neurons are devoted to vision than to any other sensory function (Zigmond *et al.*, 1999). With approximately 10^6 neurons, the insect brain is much simpler than that of primates, who have roughly 10^{11} neurons. This is also true for the insect visual system as compared to the primate visual system. A simple organization of the insect brain showing prominent layers of the visual system is shown in Figure 1.1. Most insects have compound eyes with multiple light gathering units known as *ommatidia*. In flies, each ommatidium consists of a lens that forms an image onto the tip of the *rhabdom*, a light guiding structure in the *photoreceptor cells* containing photo pigments (Land and Fernald, 1992). This visual information is transduced into electrical signals by the photoreceptor cells at the level of the retina. This transduced signal from the retina is processed by three layers of the visual ganglia: the *lamina*, the *medulla*, and the *lobula complex*. In flies, the lobula complex is further subdivided into the *lobula* and the *lobula plate*. Lobula plate neurons have been studied extensively and have been implicated in directional motion detection (Hassenstein and Reichardt, 1956; Reichardt, 1961; Egelhaaf and Borst, 1993; Borst and Egelhaaf, 1989), small-field target tracking (Land and Collett, 1974; Reichardt *et al.*, 1989; Egelhaaf, 1985a, 1985b, 1985c, 1990) and collision avoidance (Borst

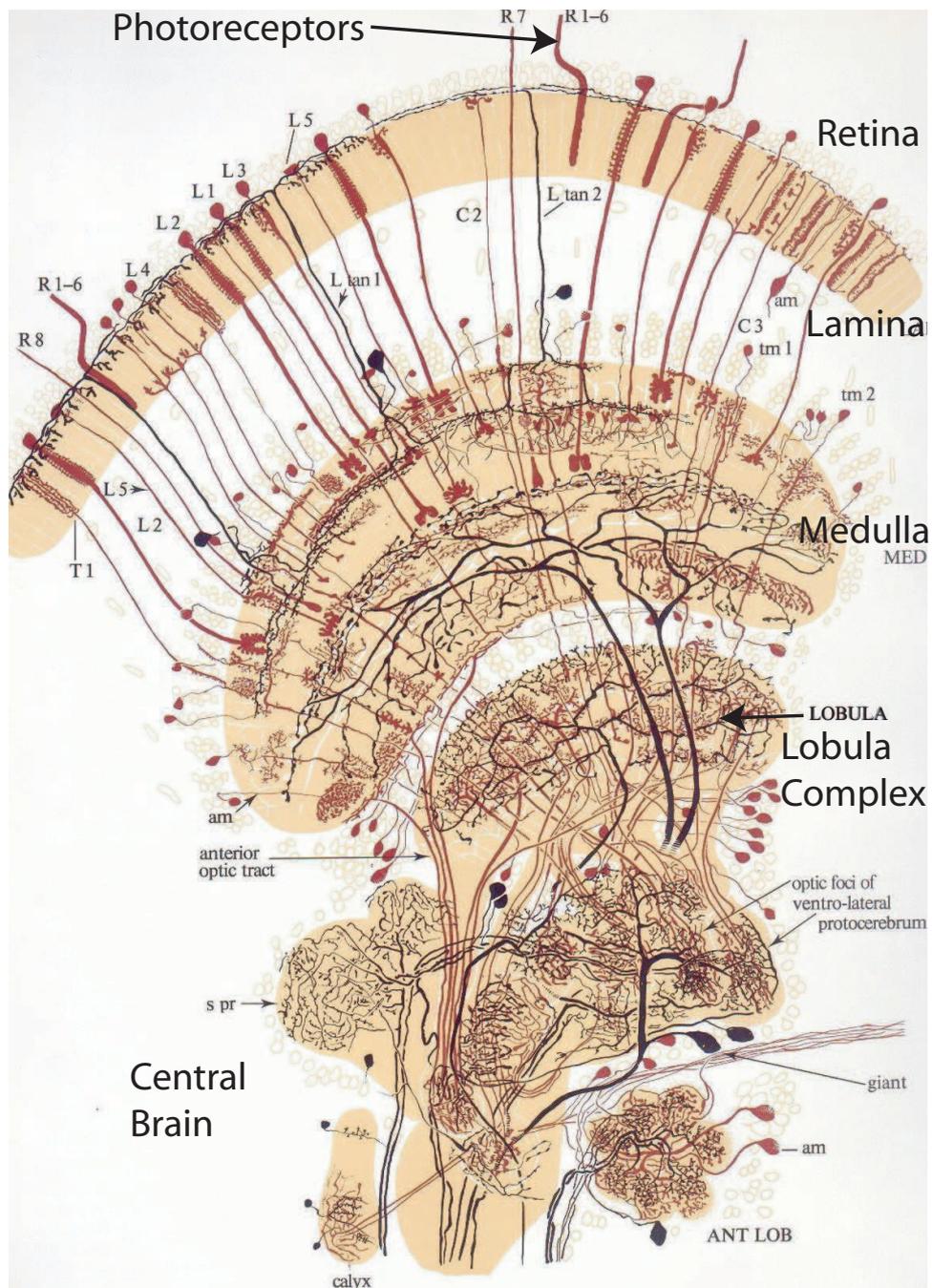


FIGURE 1.1. Diagram of the fly visual system. The outermost layer of the insect eye, the retina, has photoreceptor cells. This layer is followed by three synaptic regions: the lamina, the medulla, and the lobula complex which comprises the lobula and the lobula plate (not shown). This is followed by the central brain of the insect. The fly nervous system has roughly 10^6 neurons. Reproduced without permission from Strausfeld (1976).

and Bahde, 1986; Tammero and Dickinson, 2002a, 2002b). A recent neuronally based computational model suggests that non-directional motion may be detected earlier in the visual pathway, at the level of the medulla, than directional motion (Higgins *et al.*, 2004). This non-directional motion has speed dependent response for a range of spatial frequencies and may subserve the speed dependent behavior of the insects (Higgins, 2004).

Several computational models have been proposed to explain how insects extract directional motion information from their surroundings. These schemes have been used to model the response of cells in the lobula and the lobula plate that respond to motion in a specific direction. We have broadly classified these models under three categories detailed below.

1.1.1 Optical-flow based models

Gibson (1950) first postulated that an *optic array* is used in the “steering of locomotion”. An optic array was defined as a brightness pattern or texture incident on the eye of an observer. A continuously changing optic array was proposed to give rise to an *optical flow field* (Lee, 1980). The optical flow is mathematically defined as the instantaneous velocity of the brightness pattern at each point in an image (Horn and Schunck, 1981). However, optical flow cannot be computed at each point in the image independent of its neighboring points without additional constraints. This is because the velocity of a point in a plane has two degrees of freedom, whereas the change in the image brightness (being a scalar) has only one degree of freedom (Horn and Schunck, 1981). This problem may be overcome by adding constraints to the original problem, such as assuming that the total image brightness remains constant as the image moves. This allows for a unique solution to be found for the velocities at each point in the image.

There are some difficulties with this methodology. It has been shown that the optical flow computed on the image may not match the actual motion for non-uniform illumination (Horn, 1986). Verri and Poggio (1989) have further shown that the actual motion field and the optical flow field generally do not match. They match under very specific illumination conditions and only for smooth objects under translatory motion. Nevertheless, it has been argued by the same authors that meaningful information about the actual motion may be interpreted from the qualitative properties of the optical flow. Such qualitative properties may be found by associating the optical flow with a dynamical system. The fixed points or singularities of a dynamical system may then be used to extract useful motion properties from the optical flow, like the focus of expansion (Poggio and Verri, 1987; Verri *et al.*, 1990).

Figure 1.2 shows typical optical flows experienced by an observer under translatory and rotatory motion. In the case of translation (Figure 1.2a), the observer sees an expanding flow field with a singularity (S) along the axis of translation. Thus, without regard to the quantitative values of the flow field, an expansive optical flow suggests a translatory motion towards the singularity, which denotes the focus of expansion. Figure 1.2b shows the optical flow field generated during rotation. Here, no singular point exists in the view field of the observer, hence the horizontal flow field. Parallel motion vectors qualitatively suggest that the observer is rotating with respect to the surrounding in a direction opposite to the direction of the flow field. Similarly, a discontinuity in the optical flow along a contour suggests the presence of an object and may be used to distinguish a target from its background. Several authors have suggested that living organisms, including insects, use this information for navigating in complex and cluttered environments (Prazdny, 1980; Adiv, 1985; Warren and Hannon, 1988; Poggio *et al.*, 1989; Bulthoff *et al.*, 1989; Egelhaaf and Borst, 1993).

1.1.2 Correlation-based models

Exner (1894) first proposed the idea of a correlation-based motion detector to explain the vivid impression of apparent motion in humans by the sequential flashing of two neighboring stationary light spots. In 1956 Hassenstein and Reichardt developed a correlation-based model to explain the

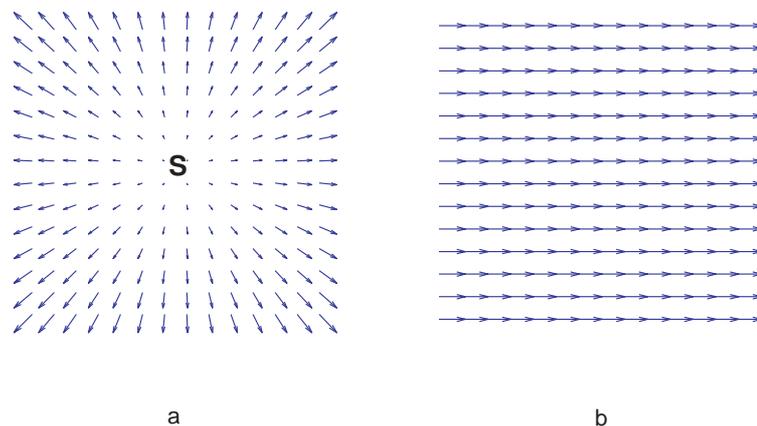


FIGURE 1.2. Optical flow fields. (a) The optical flow field generated due to translation. S denotes the location of the singularity. (b) The optical flow field generated due to rotation.

rotatory large-field motion response of the beetle *Chlorophanus* to a rotating visual stimulus. This large-field motion response is often termed the *optomotor response*. The Hassenstein-Reichardt (HR) model proposes that the two dimensional visual field is covered by local motion detectors, also known as *elementary motion detectors* (EMD), each evaluating a *spatio-temporal* cross-correlation between the light intensity fluctuations between two neighboring photoreceptors. The response of a spatio-temporal frequency tuned model, like the HR model, is dependent on both the spatial frequency and the temporal frequency components of the visual stimulus being used to elicit the response.

The HR model with an additional temporal high-pass filter is shown in Figure 1.3 (Higgins *et al.*, 2004). Each motion detector receives its inputs from photodetector units followed by two temporal filters and a multiplication unit, arranged in a mirror-symmetric fashion. The two temporal filters successively perform a high-pass and a low-pass filtering operation. The high-pass filter enhances the sharp temporal features of the image. The low-pass filter acts as a delay. The delayed and undelayed responses from two adjacent photoreceptors are multiplied to perform a correlation between the two. The difference between two such adjacent multiplication units is spatio-temporal frequency tuned and is directionally selective along the orientation of the photoreceptor axis. Electrophysiological data from wide-field lobula plate neurons in the insect visual system also known as *tangential cells* matches the response of an HR model (Reichardt *et al.*, 1983; Egelhaaf, 1985a; Egelhaaf and Borst, 1993; Egelhaaf *et al.*, 1993).

The HR model has been elaborated by adding temporal and spatial filters before the correlation operation (van Santen and Sperling, 1985). The addition of the spatial and temporal filters makes the response of this model detect motion even in the presence of noise. Adelson and Bergen (1983) have proposed a spatio-temporal energy based model that is mathematically equivalent to the elaborated

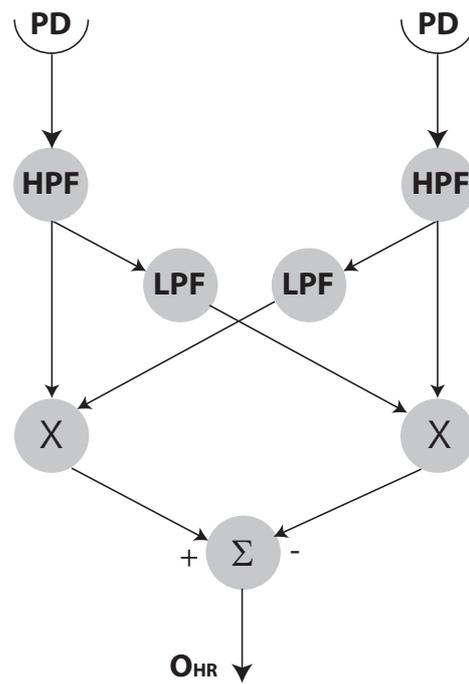


FIGURE 1.3. The Hassenstein-Reichardt (HR) model. Visual input is filtered by high-pass filter (HPF) units to remove all sustained responses from the photoreceptor (Higgins *et al.*, 2004). This response is correlated (\times) with a delayed response (low-pass filter (LPF) unit acts as a delay) from the adjacent photoreceptor output. The responses from two adjacent correlators are subtracted (Σ), and this output is direction selective (O_{HR}) such that its response magnitude is positive for stimulus motion in one direction and negative in the other.

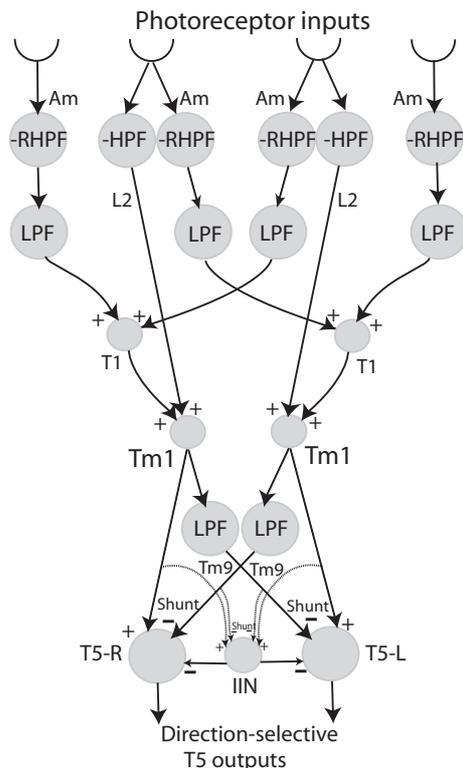


FIGURE 1.4. Neuronally based elaborated EMD model. RHPF unit indicates relaxed high-pass filter, HPF a high-pass filter, and LPF a low-pass filter. S represents a summation. Identified neurons are indicated at each processing stage. See text for details. Reproduced without permission from Higgins *et al.* (2004).

HR model. This system has been used to model the motion response of a primate cortical complex cell. Verri and Poggio (1989) have argued that biological organisms may use spatio-temporal based models instead of the optical flow fields to estimate the qualitative properties of the actual motion. Based on the above models, several schemes have been proposed to explain complex navigational strategies used by insects (Virsik and Reichardt, 1976; Reichardt *et al.*, 1983; Borst and Bahde, 1986; Reichardt *et al.*, 1989; Tammero and Dickinson, 2002a; Higgins and Pant, 2004b).

1.1.3 Neuronally based EMD model

A neuronally based small-field elementary motion detector model was recently proposed by Higgins *et al.* (2004). This model is based on a subset of neurons from the photoreceptors through the lamina (amacrine (Am), monopolar L2, and T1 cells), the medulla (transmedullary Tm1 and Tm9 cells), and the lobula (inhibitory interneurons and T5 cells) in the fly brain. Figure 1.4 shows the various computational units in the model along with identified neurons implicated for those specific computations. A photoreceptor converts incident light into signals that are processed by amacrine and lamina monopolar cells. The synapse between the amacrine and T1 cell is modeled as a sign-inverting ‘relaxed’ high-pass filter (RHPF) (Higgins *et al.*, 2004) that retains a portion of the sustained response while high-pass filtering the input signal. The activity of an L2 cell is implemented as a sign-inverting high-pass filter (HPF). The response of the T1 cell is modeled as a sum of two low-pass filtered (LPF) amacrine cells. A Tm1 cell sums the response from the L2

and the T1 cells. A T5 cell sums the response of the Tm1 cell with a sign-inverted response of a Tm9 cell that is modeled as a low-pass filter (LPF) unit. The activities of two adjacent T5 cells are mutually inhibited by the inhibitory interneuron. The output of the T5 cell is directional and feeds to the lobula plate tangential cells.

The neuronally based EMD model is consistent with the HR model and may be used to explain electrophysiological data from lobula plate tangential cells (Rivera-Alvidrez, 2005). However, there are some major differences between the two models. While the HR model does not specify how the correlation-based model is actually implemented in the brain, the neuronally based EMD model specifies anatomical equivalents to the different subunits. Also, the response of the HR model is dependent on the spatio-temporal frequency and not the speed of the object in its visual field. The response of the Tm1 unit in the neuronally based EMD model, on the other hand, is speed dependent for a range of spatial frequencies (Rivera-Alvidrez, 2005). It has been reported that honeybees use image speed to center themselves while flying through a tunnel or landing on a flat surface (Srinivasan *et al.*, 1991, 2000). There is also evidence that they may be using non-directional motion during navigation (Srinivasan *et al.*, 1993). The neuronally based EMD model may implicitly compute a non-directional motion output that is also speed dependent (Higgins, 2004).

1.1.4 Contrast saturation

The biological motion detection models described for the fly brain suffer from a common problem. Their response magnitude is dependent on the contrast of the visual input. For the correlation-based models, the response is dependent on the square of the contrast. For the neuronally based EMD model, the response magnitude is directly proportional to the stimulus contrast. Insects solve this problem by employing *contrast saturation* in the visual system. This mechanism acts so that the response of the insect visual system is dependent on the contrast of the visual input for a very small range, outside of which it saturates. This has been incorporated into elaborated models based on the correlation and neuronally based EMD model (Harris *et al.*, 2000; Rivera-Alvidrez and Higgins, 2005).

All three of the above described motion detection schemes are often used to model the visual navigation behavior of insects. In the next section, we present a brief account of these models and their robotic implementations.

1.2 Visual navigation

Visual navigation requires an interaction between visual and motor systems. The visual information from a scene needs to be parsed by the insect brain to extract structural information about its surroundings. This information must then be used by the motor system of the insect to navigate in its environment. Several navigational schemes have been suggested based on optical flow information (Warren and Hannon, 1988; Mallot *et al.*, 1991; Krapp and Hengstenberg, 1996). In addition, schemes have been proposed employing lobula plate tangential cells. It has been suggested that the vertical system (VS) neurons in the lobula plate of the fly act as *matched filters*. One navigation scheme based on the VS neurons matches the optical flow field experienced by the fly with stereotypical translatory and rotatory optical flows to compensate for self motion (Franz and Krapp, 1998). Navigational subsystems have also been proposed that utilize correlation-based motion models. Reichardt *et al.* (1989) have developed a computational model based on the HR model to describe how insects distinguish between an object of interest and the surround. Borst and Bahde (1986) have described an HR model based scheme to explain the landing behavior of flies. This model has been elaborated by Tammero and Dickinson (2002b) to explain landing and avoidance response in flies.

Some obstacle avoidance and escape response models have also been proposed that do not utilize motion. Rind and Simmons (1992) have developed a neural network based on the response of the Lobula Giant Movement Detector (LGMD) and the Descending Contralateral Movement Detector

(DCMD) neurons in the locust brain. This non-directional model utilizes edge detectors, and responds strongly to approaching objects that trigger an escape response in the locust. Laurent and Gabbiani (1998) have described a competing collision detection model based on the same neurons. They have also argued that the same model may explain the response of neurons in the *nucleus rotundus* of the pigeon brain. Another set of navigational schemes based on non-directional motion have been proposed by Srinivasan *et al.* (1993). They have shown that honeybees do not use directional motion information while flying in the center of a tunnel, but rather use non-directional image speed information. The same has been shown for distance calculation (odometry) and landing behavior in bees (Srinivasan *et al.*, 1996, 2000). It has also been reported that two distinct pathways, one directional and the other non-directional, may be behind the optomotor and the image-speed computation, respectively (Srinivasan *et al.*, 1993). The HR model is widely used to represent the pathway that exhibits the optomotor response. However, the response of this model is dependent on the spatial and temporal frequency, and not the speed of the stimulus (Srinivasan *et al.*, 1993). A non-directional model that is based on the transmedullary Tm1 cells in flies (Higgins, 2004; Rivera-Alvidrez, 2005) may represent the pathway that computes image speed. This model may therefore be used to explain the centering, odometry, and landing behavior of the honeybees (Higgins, 2004; Rivera-Alvidrez, 2005).

Robotic implementations of the biological models based on optical flow have been devised and shown to be successful in simple scenarios (Weber *et al.*, 1997; Verschure, 1998; Huber and Bülthoff, 2002). LGMD neuron based robotic systems (Blanchard *et al.*, 2000; Santer *et al.*, 2004; Stafford *et al.*, 2007b) and those based on the HR model (Huber and Bülthoff, 1998; Harrison, 2005) have also been executed with limited success.

A parallel effort has been going on in the computer vision community to come up with ‘intelligent’ navigational schemes not based on biological models. For collision detection, several optical flow based algorithms have been described (Lee, 1980; Ancona and Poggio, 1995). It has been argued that if the observer is *active* (a sensor that can adjust its orientation between two frames) as opposed to *passive* (a fixed sensor), the problem of extracting motion information from the optical flow becomes much simpler (Bajcsy, 1985; Adiv, 1985; Aloimonos *et al.*, 1988; Bandopadhyay and Ballard, 1991). Mathematical models have been proposed for binocular observers which utilize tracking (Aloimonos *et al.*, 1988). However, they require exact computation of several parameters based on the motion of the object and may not be usable for real-time systems. Monocular observer based algorithms have also been proposed (Fermuller and Aloimonos, 1992). However, the implementation of such mathematical algorithms for an active viewer have only been shown for very simple and controlled scenarios (Fermuller and Aloimonos, 1993). In this dissertation, we focus on a simple monocular agent which can detect collision in non-trivial scenarios.

In the next section, we present a brief account of the biologically inspired artificial sensors that have been implemented as VLSI architectures.

1.3 Neuromorphic Engineering

Neuromorphic Engineering applies the design strategies of neurobiological systems to electrical engineering. Inspiration is taken from neurobiological circuits and models which are implemented through analog building blocks and novel circuits to design ‘intelligent’ systems. Mead (1989) pioneered the field by implementing neural circuits in silicon hardware and one of the first examples was an analog circuit that mimicked a spiking neuron (Mahowald and Mead, 1991). Visual, auditory, and other bio-sensory models have since been successfully implemented (Andreou *et al.*, 1991; Mahowald and Mead, 1991; Horiuchi *et al.*, 1991; Sarpeshkar *et al.*, 1993; Delbruck, 1993; Etienne-Cummings *et al.*, 1993; Lazzaro *et al.*, 1993; DeWeerth and Morris, 1994; Kramer *et al.*, 1995; Liu, 1996; Lazzaro and Wawrzynek, 1997; Harrison and Koch, 1998; Kumar *et al.*, 1998). Visual motion based models have been the most widely implemented sensory models. As these

systems became more complex, the computation was divided among several chips to implement modular architectures (Higgins and Koch, 1999; Indiveri *et al.*, 1999; Higgins and Shams, 2002; Ozalevli and Higgins, 2005). Several authors have implemented a biologically inspired asynchronous communication protocol to communicate between chips in a multi-chip system (Boahen, 1998; Kalayjian and Andreou, 1997; Landolt *et al.*, 2001). A thorough analysis of the fundamental issues encountered in the design of motion processing architectures is reported in Sarpeshkar *et al.* (1996). Higgins *et al.* (2005) have presented a comparative study of three different motion detection models under similar test conditions. Biomimetic computational architectures have also been successfully implemented as analog VLSI chips for target tracking, time-to-collision computation, and collision detection (Etienne-Cummings *et al.*, 1996; Indiveri *et al.*, 1996, 2002; Miller and Barrows, 1999; Horiuchi and Niebur, 1999; Etienne-Cummings *et al.*, 2000; Higgins and Pant, 2004a; Harrison, 2005). The particular attraction of the analog circuit architecture is that it can be used to implement a neuronal system in continuous time, the way processing actually happens inside a neuron. At the same time, an analog chip implemented in the subthreshold regime of the MOSFET consumes little electrical power. Therefore, it is well suited to be used as a sensor in an autonomous system.

In the next section, we present an outline of the work that we will describe in this dissertation.

1.4 Summary of the dissertation

In this dissertation, we present biomimetic models that may be used in the design of intelligent systems. The biological models have an advantage over their non-biological counterparts. They present elegant and compact design methodology as compared to complicated and computation-intensive computer algorithms. We describe the problem of autonomous navigation, specifically focusing on biological collision avoidance models. We present a comprehensive simulation study of two representative biological collision detection models in Chapter 2. The two models have been chosen for their simple design that is suitable for implementation as a real-time artificial system. In Chapter 3, we mathematically analyze these models and present elaborations to these models based on the idea of active vision. Fermuller and Aloimonos (1992) have shown that tracking may make the computation of time-to-collision feasible with only the optical flow information. However, due to the inherent problems in computing optical flow this algorithm may not be able to detect collision under real world scenarios. Correlation-based models, on the other hand, may be used to compute spatio-temporal frequency tuned responses in a precise manner. These responses are also qualitatively similar to the optical flow field (Verri and Poggio, 1989) and therefore, correlation-based models may be used instead of the optical-flow based models. The biological collision detection models described in Chapter 3 are based on spatio-temporal frequency based models. We use the tracking-based approach to improve the collision detection models and present simulation results of these elaborated models. In Chapter 4, we present a physical implementation of the two tracking-based collision detection models. We present the results of this system for collision and non-collision scenarios and compare the two collision detection models based on their performances.

Collision avoidance may also be achieved by determining the distance of an object from the observer. This may be accomplished by utilizing relative speed of objects in a visual scene to calculate relative depth. Navigation in honeybees has been argued to be controlled by a similar image speed computation (Srinivasan *et al.*, 1991, 1996; Zhang *et al.*, 1993). In Chapter 5, we simplify two non-directional motion models derived from the neuronally based EMD model whose response is speed dependent, and present their simulation results. The speed response of these models are independent of a range of spatial frequencies of the visual stimulus. Based on these models, we present the design of an analog VLSI sensor chip and its characterization results in Chapter 6. This non-directional speed sensor may be utilized in estimating the relative depth of an object and may be used in the design of autonomously navigating systems. Finally, in Chapter 7, we discuss the performance issues related to our work and present ideas for future work.

CHAPTER 2

A COMPARATIVE STUDY OF BIOLOGICAL COLLISION AVOIDANCE
MODELS

Biological organisms from insects to primates display a stereotypical *escape response* when an object in their visual field expands rapidly, a phenomenon termed ‘looming’. This response must be distinguished from an obstacle avoidance response. The escape response is a reflexive reaction to an imminent danger, while an *obstacle avoidance* response is a navigational strategy used under normal circumstances to avoid direct collisions (Reynolds, 1999; Cuadri *et al.*, 2005a). In biological systems, the escape response is often mediated by a direct path between the sensory and the motor system (Rind and Simmons, 1992; Laurent and Gabbiani, 1998). Visual obstacle avoidance, on the other hand, is a computationally involved process based on optical flow information that may lead to a variety of behaviors (Land and Collett, 1974; Reichardt *et al.*, 1989; Srinivasan and Zhang, 1997). In our effort to design an intelligent system that may autonomously navigate in its environment, we perform simulation studies of both the escape and obstacle avoidance models described for the insect visual system. Obstacle avoidance based navigational models are discussed in Chapters 5 and 6. In the next few chapters, we simulate the escape response based models that have been proposed for collision detection for collision and non-collision scenarios. We term these models *collision avoidance models* and present their details in the following section.

2.1 Collision avoidance models

We have focused on three leading neuronal models hypothesized to underlie collision avoidance. The first model that we have considered is based on the vertical system (VS) neurons in the fly brain. These neurons have been hypothesized to supply visual cues for the control of head orientation, body posture, and flight steering (Borst, 1990; Krapp *et al.*, 1998). This scheme is based on a matched filter approach (Franz and Krapp, 1998), whereby motion information is filtered by a set of neurons to extract information for specific tasks, including collision avoidance. A matched filter based collision avoidance model was recently proposed by Tammero and Dickinson (2002a) based on their behavioral studies of fruit flies. We will call this model the Spatial and Temporal Integration (STI) based model and describe it in Section 2.2.

The other two models that we describe are based on two of the most studied neurons mediating collision avoidance in insects. These are the Lobula Giant Movement Detector (LGMD) and the Descending Contralateral Movement Detector (DCMD) neurons (Rind, 1984; Milde and Strausfeld, 1990; Rind and Simmons, 1992; Hatsopoulos *et al.*, 1995). Rind and Bramwell (1996) have modeled the response of the LGMD neuron using a neural network. We refer to this model as the Rind model and describe it in Section 2.3.1. A competing mathematical model based on the response of the same two neurons has been described by Laurent and Gabbiani (1998). We refer to this model as the η -function based mathematical model, which is discussed in Section 2.3.2.

2.2 The spatial and temporal integration based model

A model for collision avoidance and landing response in the fruit fly *Drosophila melanogaster* was recently proposed by Tammero and Dickinson (2002a). This model is an elaboration of a scheme based on the spatial and temporal integration of small-field motion units, proposed by Borst and Bahde (1986), to explain the landing response of the housefly. Borst and Bahde studied the stereotypical leg extension response of the fly while presenting an expanding visual stimulus. They concluded

that when the expansion output exceeds a threshold, the fly extends its legs to land. Tammero and Dickinson have elaborated this model to account for both landing and collision avoidance. As shown in Figure 2.1, the STI model consists of an array of Hassenstein-Reichardt (HR) type correlation-based motion detectors (Reichardt, 1961) that compute a qualitative representation of the optical flow field (Verri and Poggio, 1989). The visual field is separated into three sub-fields: left, center, and right. In a two dimensional (2D) version of this model, the outputs from individual HR detectors in each sub-field are filtered into rightward, leftward, upward, and downward motion components. These motion components are then spatially combined such that the output in each of the four quadrants represents a net outward motion or expansion (see Figure 2.1ii). These expansion components are temporally integrated by a leaky-integrator unit. In the left and right sub-fields, the expansion on one side (say, left) is subtracted from the expansion on the other side (right). In the central sub-field, the expansion output remains unchanged. The output in each sub-field is compared with a threshold to generate either a landing response (based on the response of the central sub-field), or an avoidance response (based on the responses of the left and right sub-fields). If the response on the right side is greater than the left side, the fly turns leftwards and vice versa. The expansion response that triggers a landing response in the fly may also be used to detect a direct collision. Harrison (2005) has recently adapted this model into an analog VLSI chip showing reasonable performance. A robot mounted with two such VLSI sensors was also designed and the performance of this robot was tested for navigational tasks. It was shown that the robot was able to navigate with few collisions. However, when and why the collisions occurred was not elaborated. In this chapter, we investigate the STI model by presenting it with real-world videos of collision and non-collision scenarios to explore the shortcomings of the model. Since this model has been extensively tested for artificial stimuli and real-world videos (Harrison, 2005), we have not duplicated those results in this chapter. Here, the model has been simulated for the most complicated videos of collision and non-collision scenarios and the results have been compared with other schemes.

2.3 LGMD-DCMD neurons based models

In this section, we describe the two models based on the LGMD-DCMD neurons in the locust brain.

2.3.1 Rind et al's LGMD based neural network

Rind and Bramwell (1996) proposed an LGMD-DCMD neuron based model for collision avoidance in a locust. The DCMD neuron has a one-to-one synaptic connection with the LGMD neuron such that a spike in the LGMD elicits a spike in the DCMD (Rind, 1984). The responses of these neurons have been well characterized (Simmons and Rind, 1992; Hatsopoulos *et al.*, 1995). Intracellular electrophysiological data recorded from the DCMD neuron show that it responds strongly to expanding stimuli, but not to sustained motion. Rind and Simmons (1992) recorded extracellularly from the DCMD neuron and reported that it also responds to novel initiation of motion, but not to slow lateral motion. Based on these recordings, Rind and Bramwell (1996) proposed a four layered neural network that mimics the biological data very closely (Figure 2.2a). The first layer maps the visual scene to an array of photoreceptors (**P** units). The photoreceptors have been assumed to be edge detectors. Each photoreceptor responds to a change in the illumination level in its visual field as an ON/OFF response and passes this activity to a second layer. This layer has an excitatory (**E**) unit and an inhibitory (**I**) unit. The **E** and **I** units are followed by a third layer of summation (**S**) units. An **S** unit receives excitatory input from the **E** unit in the same retinotopic location, and inhibitory inputs from **I** units of its next and next-to-next neighbors. In all, an **S** unit has twelve inhibitory inputs (see Figure 2.2b) and one excitatory input. The input from the **I** units are weighted according to their relative position with respect to the **S** unit. The nearest six **I** units are weighted by a factor of 1/6, while the further six units are weighted by a factor of 1/12. Time delays of Δ and 2Δ are associated with the nearest and next-nearest **I** units. All the **S** units project

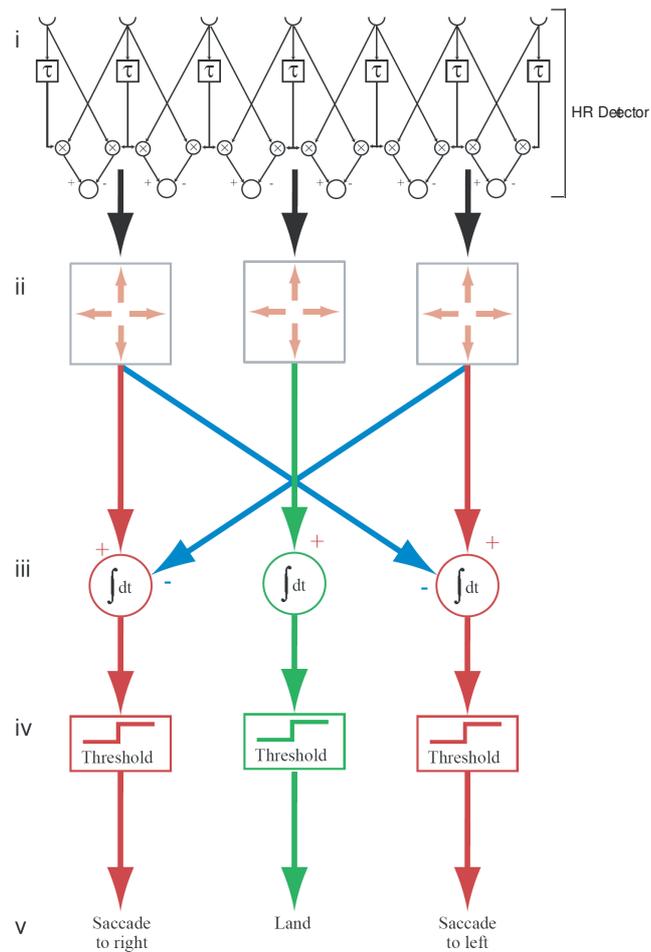
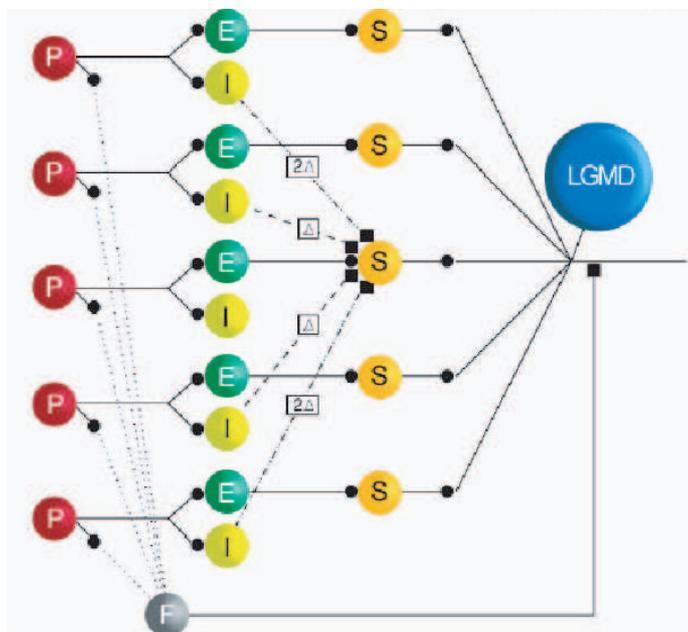
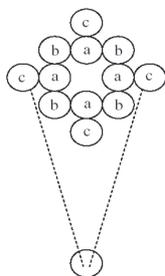


FIGURE 2.1. Tammero and Dickinson's model for collision avoidance and landing response. (i) The two dimensional motion field is computed by correlation-type (HR) motion detectors. (ii) Spatial aggregation of this local information is performed in the leftward, central and rightward regions of the visual field. (iii) This output is temporally integrated to determine the global motion as seen by a fly. (iv and v) The output is compared with a threshold and for lateral motion, a collision avoidance maneuver is elicited. If the expansion is in the central region of the visual field, a landing response is generated. Figure reproduced without permission from Tammero and Dickinson (2002a).



a)



b)

FIGURE 2.2. Rind and Bramwell's neural network architecture. (a) The four processing layers of the neural network in 1D. **P** units represent the photo-transduction layer, followed by an excitatory and an inhibitory layer (**E** and **I** units). **S** units aggregate response from **E** and lateral **I** units forming the third layer of the network. All the **S** units project upon an **LGMD** unit, the fourth layer of the network. The **LGMD** unit also receives a feed-forward inhibition input from an **F** unit which gets activated when a certain number (~ 50) of **P** units respond together. (b) 2D connections between 12 neighboring **I** units and a **S** unit. Reproduced without permission from Cuadri *et al.* (2005a).

upon an **LGMD** unit that forms the fourth layer of this network. The **LGMD** unit also receives an inhibitory input from a single **F** unit. The **F** unit constitutes a feed-forward inhibition pathway and is triggered when a sudden burst of activity is detected by a fixed (~ 50) number of **P** units. This prevents a sudden change in the illumination from triggering a collision avoidance response. The **LGMD** unit computes the sum of all the **S** units, and fires when this sum exceeds a threshold. A race between the central excitatory and lateral inhibitory inputs dictates whether or not the **LGMD** unit will trigger a collision avoidance response. Since the computations in this neural network are performed locally in the **S** units, it may not distinguish between an approaching object and a fast horizontally translating object as long as the sum of the **S** units exceeds the threshold. It has been suggested that this is not a problem in naturalistic scenarios. In nature, a horizontally translating object usually expands at a much slower rate on the retina compared to an approaching obstacle. Thus, the horizontal motion would elicit a much smaller response from the **E** units (Cuadri *et al.*, 2005a). The Rind model relies on the lateral inhibition from the neighboring **I** units to suppress this excitatory input from triggering the **S** unit. However, in the case of a collision scenario the expansion is so rapid that the lateral inhibition network loses the race to the **E** units and the **LGMD** unit is able to trigger a collision avoidance response. The delays between the **I** and the **S** units are critical in this respect. They can be optimally chosen such that the **LGMD** unit is able to detect a collision scenario from a horizontal translation. All the layers in this model also have a short-term memory (persistence) associated with them. Santer *et al.* (2004) have shown that this persistence is critical to the performance of this model. We have simulated three different versions of the Rind model in search of a reliable collision avoidance scheme and we present their details in Section 2.4.2.

2.3.2 The η -function based mathematical model

A competing model, also based on the LGMD and DCMD neurons, was proposed by Hatsopoulos *et al.* (1995). Unlike what was reported by Rind and Simmons (1992), the data recorded by Hatsopoulos *et al.* suggests that the response of the DCMD neuron peaks and then falls off at a fixed delay after the angle subtended by a looming object has exceeded a threshold. This response is invariant to the actual size, speed, shape, or angle of approach of the object (Gabbiani *et al.*, 2001). This model has been elaborated by Laurent and Gabbiani (1998) to also explain data from the *nucleus rotundus*, an area in the brain of a pigeon (Sun and Frost, 1998). This model utilizes the near-exponential growth in the size of an approaching obstacle to generate a collision avoidance response. In a one-dimensional (1D) case, the response of the LGMD and DCMD neurons to an approaching object of size d subtending an angle $\theta(t)$ and moving at a speed v has been formulated in terms of the η -function (see Figure 2.3a). This model defines $t < 0$ as time before collision and $t=0$ as the time of expected collision. The angle subtended by the object on the retinal plane can be computed by simple trigonometric calculation to be:

$$\theta(t) = 2 \tan^{-1} \left(\frac{d/2}{v \cdot t} \right) \quad (2.1)$$

An approaching object's angular size and its rate of change grows near-exponentially on the retinal plane (see Figure 2.3b). Taking the derivative of $\theta(t)$, we get:

$$\dot{\theta}(t) = 2 \cdot \frac{d/2v}{t^2 + (d/2v)^2} \quad (2.2)$$

This rate of angular expansion along with the angular size are hypothesized to be controlling the DCMD activity. This activity of the DCMD neuron is modeled as:

$$\eta(t) = \frac{\dot{\theta}(t)}{e^{\alpha \cdot \theta(t)}} \quad (2.3)$$

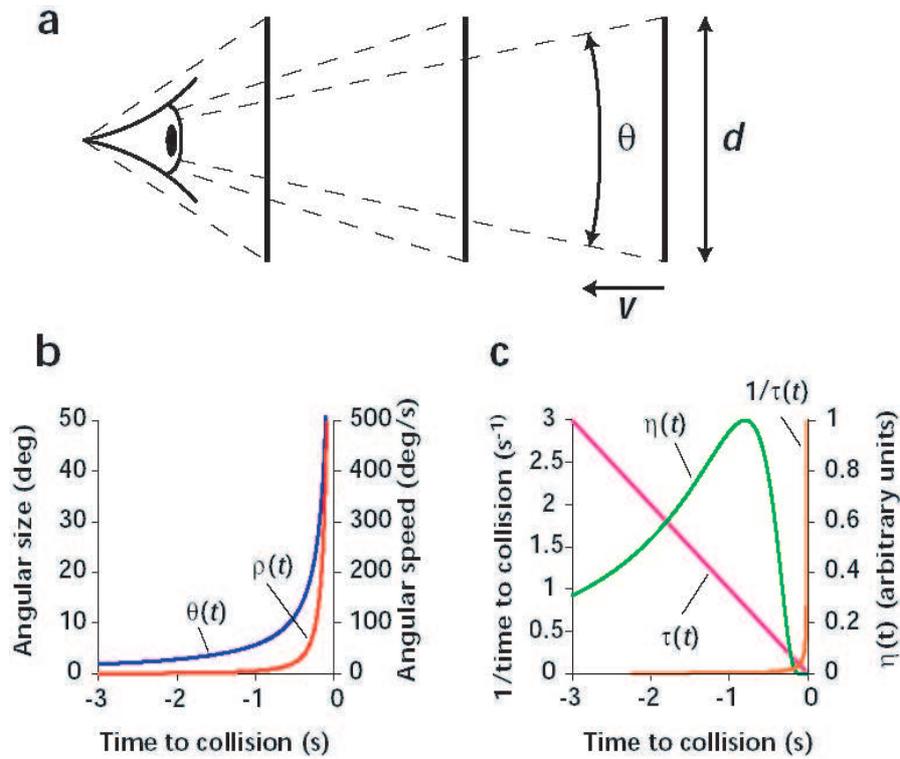


FIGURE 2.3. The η -function based computational model. (a) An object of size d approaching at a speed v subtends an angle θ upon the eye. (b) Time course of angle $\theta(t)$ and its rate of change $\rho(t)$ during approach. (c) Time course of $\eta(t)$, $\tau(t)$ and $1/\tau(t)$ during approach. Reproduced without permission from Laurent and Gabbiani (1998).

where α is a parameter that sets the threshold at which the angular size of the object starts to inhibit the DCMD neuron. The angular expansion builds up as the object approaches, increasing the response of the DCMD neuron along with it. However, when the object's angular extent exceeds a threshold (θ_{thr}), the denominator of the η -function causes the neuron's firing rate to peak and then decay exponentially. α is modeled by the following equation:

$$\tan\left(\frac{\theta_{thr}}{2}\right) = \frac{1}{\alpha} \quad (2.4)$$

However, for this scheme to work, the insect must be able to compute the size of an approaching obstacle, and also be able to detect the peak firing rate of a neuron. Both of these calculations are non-trivial and require computational time that may not be available in collision scenarios (Laurent and Gabbiani, 1998). Also, the peak signal represents the angular size and not the actual size. Thus, a large object farther away would signal the same response as a smaller object that is nearby. This implies that less time will be available to escape from a fast-moving near object. This problem can be resolved by computing how much time is left before collision. Using this model, Laurent and Gabbiani have calculated a time-to-contact (τ) variable. As the name suggests, this variable represents the time left before a looming object will collide with the observer. Using a small angle

approximation for $\theta(t)$, the time-to-contact variable is calculated as:

$$\tau(t) \approx \frac{\theta(t)}{\dot{\theta}(t)} \quad (2.5)$$

Figure 2.3c shows the time course of the η -function and the variable $1/\tau(t)$ during an approach.

This mathematical model differs from the Rind model in many ways, even though both are based on the response of the same LGMD-DCMD neurons. The Rind model extracts edge information from the activity of the photoreceptors (**P** layer). The number of photoreceptors that get activated (size information) and how fast this activity spreads (edge expansion rate or the speed of approach) dictate the response of this model. The η -function, on the other hand, requires both the peak firing rate and the rate of change of the angular size to compute the collision avoidance response. However, the η -function based model does not describe a methodology as to how this information is available to the LGMD system. Another important distinction between the Rind model and the η -function based model is the manner in which the size of the object affects the response of the system. The η -function is dependent on the ratio of the size and speed of the object (d/v) but not its absolute size. The Rind model, however, has direct dependence on the size of the approaching object (see Section 2.5). Neuronal data recorded from the DCMD neuron by Rind and Simmons (1992) show a size dependence. However, the data from Gabbiani *et al.* (1999) suggest that the DCMD firing rate is independent of the size and depends only on the ratio of the size and speed of the approaching object. Since the η -function based model requires computing the exact size of the obstacles and their rate of change, in an engineering sense we would need to design a system capable of extracting this information from the visual scene. This is not an easy computation. The size information can be accurately estimated only by extracting the object from its background which leads us to the problem of visual segmentation. Biological visual segmentation algorithms have been proposed (Koch *et al.*, 1986) and their VLSI implementations have even been fabricated (Stocker, 2004). However, these are complex solutions and require significant processing hardware and computational time. In our quest for a simple collision avoidance system, we have therefore not pursued the η -function based model further in this dissertation.

2.4 Methods

Numerical simulations of the STI and the Rind models were performed using Matlab (The Mathworks, Inc., Natick, MA). Simulations were performed on a set of computer generated stimuli and real-world videos.

2.4.1 Visual stimuli

Visual stimuli used for testing the performance of the models can be broadly categorized into the following groups:

1. Approaching and receding computer-generated objects
2. Real-world videos of collision and non-collision scenarios

The visual arena for all the computer-generated scenarios was three-dimensional (3D). The visual field of the simulated monocular ‘insect’ was a two-dimensional (2D) spherical projection of the 3D world, with both an azimuth and elevation extent of 180° . The size of the computer-generated visual stimulus was 300×300 pixels. We averaged every five neighboring pixel values to generate a five-times downsampled visual field of size 60×60 . This was done for two reasons. Firstly, the pixel by pixel motion in the 300×300 visual field is jerky and downsampling it provides smoother motion of the objects. Secondly, this spatial blurring mimics a similar spatial smoothing in the

insect visual system (Land and Fernald, 1992). The contrast of the computer-generated stimulus (a black or a white square) could be set at the start of an experiment and was usually set to be 1 (white). As the object approached the simulated insect, its angular size increased as an inverse function of its distance from the observer, as is expected in the case of a real approach. The response and time course of the simulations were normalized to match with responses from the original Rind model such that each timestep of the simulation was equivalent to 0.46 ms, and each space unit was equivalent to 5.6 cm. The magnitude was scaled by a factor of 0.68 to match it with the excitation level of the original model at the fastest speed.

The videos used in our simulations were the same as used by Cuadri *et al.* (2005a) for testing their Rind model based collision detection algorithm. This was done so that we could compare our results with their observations. The car videos recorded cases of direct-collision, a road drive with cars passing by, and cars translating horizontally past a stationary car at a close distance. We also tested for a reverse-collision case where the actual collision movie was played frame-by-frame in reverse time. The car videos had an image size of 240×368 and were also downsampled by a factor of 5 to a size of 48×73 , to simulate spatial smoothing. This was done to make it comparable to the simulations performed using the computer-generated stimuli.

2.4.2 Computational models

In this section, we describe our implementation of the computational models that were used in this simulation study.

Continuous-time Rind model: We implemented the Rind model in the continuous-time domain by incorporating temporal filters in place of the fixed time-delays (see Figure 2.4). This novel implementation simplifies the original Rind model while retaining all its features. The edge detector based \mathbf{P} layer of the original Rind model is replaced by a phototransducing unit PR and a high-pass filter HPF. The low-pass filter units used in place of fixed delays have an added advantage. The response of the low-pass filter persists for some time based on its time-constant. This eliminates the need for artificially including persistence in the excitatory and inhibitory nodes as in the case of the original Rind model. Instead of discrete frame-by-frame processing we have ‘continuous’-time processing, the way visual information is actually processed by neurons. Our implementation of the original Rind model is referred to as the *continuous-time Rind model* from here onwards.

We simulated an array of 60×60 processing units to match the pixel size of the computer generated stimuli. In the case of real-world videos, the number of processing units was modified to 48×73 units to match the aspect ratio of the video. Each processing unit for the continuous-time Rind model (see Figure 2.4) comprised a photoreceptor unit, an excitation and inhibition unit, a summation unit, a feed-forward inhibition unit, and an LGMD unit. The luminance information from the visual stimulus was used as the input to the photoreceptors. A high-pass filter (HPF) was implemented to extract temporal edge information from the stimulus. This was done to make our model comparable to the ON/OFF response of the photoreceptors in the original Rind model (Rind and Bramwell, 1996). This edge information was passed onto the excitatory and inhibitory units. These units were implemented as low-pass filters. The time-constant of the excitatory units (LPF_E) was smaller than that of the lateral inhibitory units (LPF_I), similar to that in the case of the original Rind model. The slower lateral inhibition was designed such that the response to a laterally moving stimulus could be inhibited by these units. However, for a rapidly expanding stimulus this inhibition was slow and would not be able to inhibit the activity of the excitatory units. The low-pass filter implementation of the excitatory and inhibitory units eliminated the need to associate an explicit delay and persistence with these units as in the original Rind model. The low-pass filters were implemented by using a buffer to store past values of the input signal. The length of each buffer was five times the time-constant of the filter. The buffers were initialized to zero in Experiment 1 to model the flash response of the original Rind model (for details see Section 2.5.1). In Experiment 2,

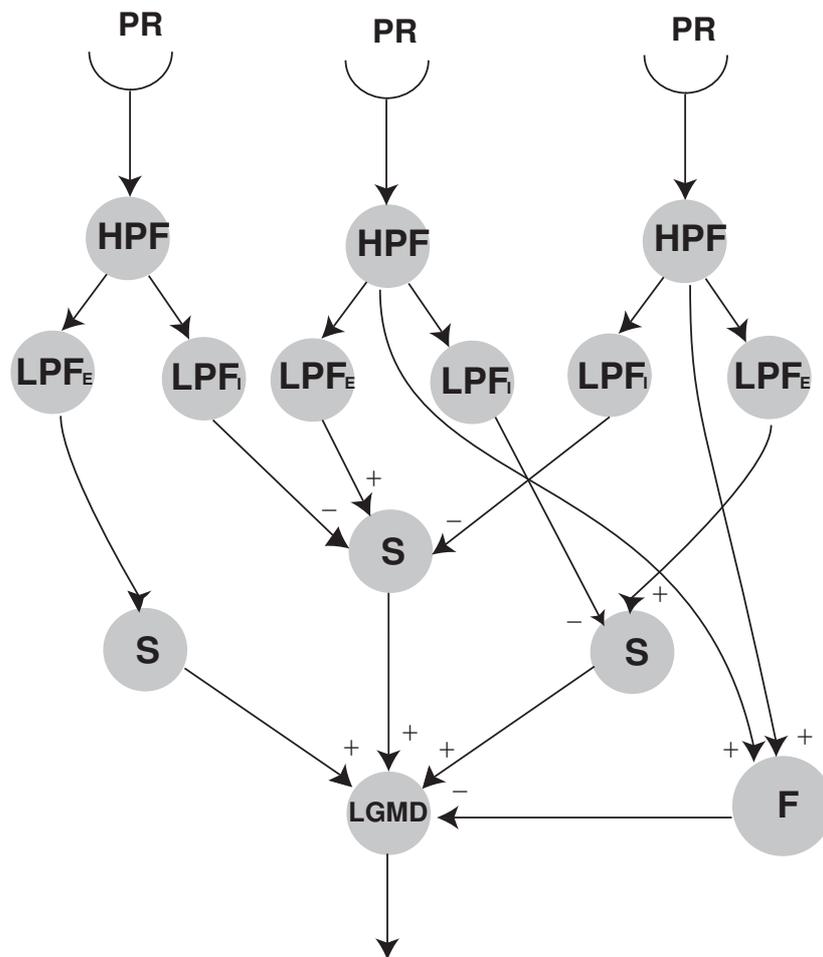


FIGURE 2.4. Continuous-time Rind model. This model is a novel implementation of the Rind model. The photoreceptor layer is a combination of phototransducing units (PR) and a high-pass filter stage (HPF) which replaces the edge detector in the original model. The second layer has two low-pass filtering units for excitatory (LPF_E) and inhibitory (LPF_I) units. The third layer is a summation layer (S) that receives positive input from the LPF_E unit and negative input from neighboring 12 LPF_I units (not all connections shown). A feed-forward inhibition unit (F) aggregates the high-pass filter output from all the photoreceptor units (not all connections shown). The fourth layer has an LGMD unit which receives positive input from all the S units and a negative input from the F unit.

buffers were initialized by running the simulation with the first frame of the stimulus for a duration equal to the longest buffer. Each summation unit (S) received input from one excitatory unit and 12 adjacent inhibitory units. Unless otherwise described, the feed-forward inhibition unit (F) was implemented as the sum of all the photoreceptor units weighted by a gain factor and was always active. The output from all the summation units and the feed-forward inhibition unit was pooled by the LGMD unit.

Continuous-time Rind model with rectification: In Experiment 2 (Section 2.5.2), we used a slightly modified version of this model. The modifications were implemented to make the model more closely resemble biological computations. We introduced a rectification stage after the excitatory (LPF_E) and inhibitory (LPF_I) units, thus making the output of the second layer strictly positive. In biology, an excitation is always positive and an inhibition is always negative. The rectifier unit ensured that the same would take place in the continuous-time Rind model with rectification as well.

Continuous-time Rind model with motion input: Stafford *et al.* (2007a) have recently implemented an altered version of the original Rind where HR unit based expansion outputs (same as the expansion computation in the STI model described below in Equation 2.6) were used after the S units. The entire visual field was subdivided into several blocks and the outputs of S units in each block were summed together. These summed responses were then correlated with adjacent blocks to compute a collision avoidance output. We have implemented a comparable model, although with some key differences. In our novel implementation, we have utilized the HR units as the first layer of processing (see Figure 2.5). This was done for two reasons. Firstly, in insects elementary motion detection, as modeled by the HR unit, takes place at the level of the lobula plate before the LGMD-DCMD neurons. Our model, therefore, may be closer to biology than the implementation of Stafford *et al.* Secondly, the output of the HR unit being motion dependent, a sudden change in illumination (flicker) does not affect its response. This makes the response of the later stages of this model more robust to flicker than the continuous-time Rind model and reduces false alarms. Up to the *Exp Filter* stage, this model is exactly similar to the STI model described below. We were able to eliminate the second order effect of feed-forward inhibition unit (F) in this model, as it was unnecessary due to it not being triggered in almost all cases. The remaining processing stages of the continuous-time Rind model were not altered. We used this model in Experiment 2 to study whether it provides an improvement over the other continuous-time Rind models (Section 2.5.2).

STI model: We also simulated a Tammero and Dickinson (2002a) type system to compare the performance of our implementation of the Rind model with the STI model. The STI model was implemented for a monocular simulated insect with a 2D sensor of angular extent 180° in both the azimuth and elevation. The input layer was an array of HR units that compute motion along two orthogonal directions of the sensor plane. The sensor plane was divided into four quadrants and the motion vectors pointing in the outward directions in each quadrant were summed together (for example, up and right motion components in the top-right quadrant). The total expansion output (E) was computed as follows:

$$\begin{aligned}
 E = & \sum_{top-left} (\sqrt{neg(M_x)^2 + pos(M_y)^2}) + \sum_{top-right} (\sqrt{pos(M_x)^2 + pos(M_y)^2}) \\
 & + \sum_{bottom-right} (\sqrt{pos(M_x)^2 + neg(M_y)^2}) + \sum_{bottom-left} (\sqrt{neg(M_x)^2 + neg(M_y)^2}) \quad (2.6)
 \end{aligned}$$

where $pos()$ and $neg()$ are functions that extract the positive and negative components of their inputs, respectively. M_x and M_y are motion outputs along the x and y directions of the 2D sensor plane, respectively. M_x is positive when the motion was left to right and negative otherwise. M_y is positive when the motion was bottom to top and negative otherwise. The model was structurally

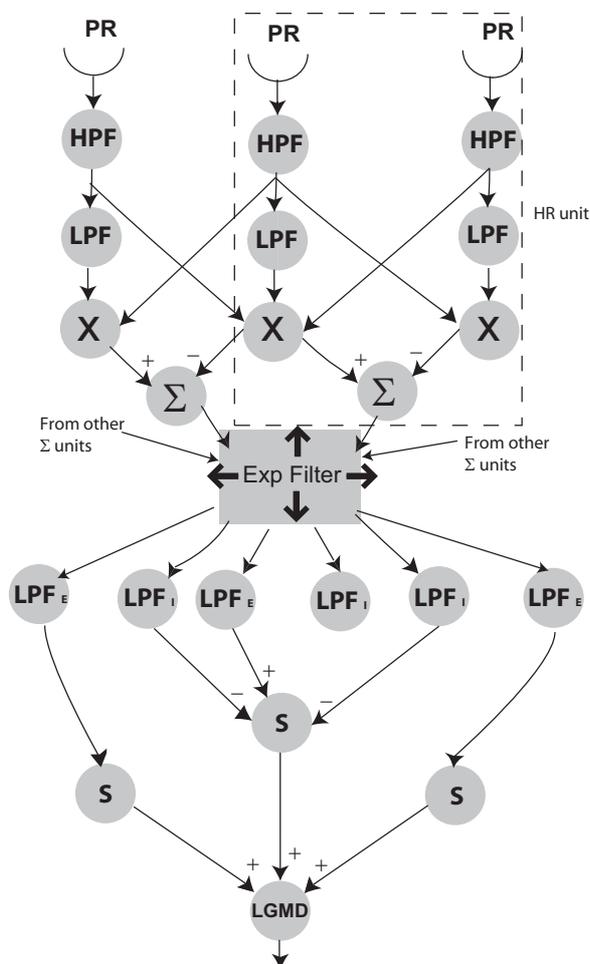


FIGURE 2.5. Continuous-time Rind model with motion input. The input layer of the Rind model has been replaced by a Hassenstein-Reichardt (HR) motion detector unit. The motion output from the HR units is processed by an expansion filter (Exp Filter). This combines motion output of HR units along the horizontal and vertical axes to generate motion components in a radially outward direction (as shown by the arrows in the Exp Filter unit). Up to this stage, the model is the same as the STI model. This expansive motion is fed to the excitatory (LPF_E) and inhibitory (LPF_I) units. The expansive and inhibitory signals are combined together in an S unit. All the S units are summed together in an LGMD unit. In this model, the F unit was not utilized.

the same as the continuous-time Rind model with motion input up to the expansion filter stage (see Figure 2.5).

This motion output is the same as the response generated by the central sub-field in the Tammero and Dickinson model (see Section 2.2). This model has earlier been simulated and tested for various collision and non-collision cases by Harrison (2005). In this chapter, we have simulated the STI model in Experiment 2 to compare its collision avoidance response with similar responses from the continuous-time Rind models (see Section 2.5.2).

2.5 Results

To compare the performance of the continuous-time Rind models with the original, we first performed experiments similar to those published by Rind and Bramwell (1996). The details of the experiments and their results are described in Sections 2.5.1. In Section 2.5.2, we present a comparison between the continuous-time Rind model with rectification, continuous-time Rind model with motion input, and the STI model based on their responses to the real-world car videos evaluated by Cuadri *et al.* (2005b).

2.5.1 Experiment 1

The original Rind model was tested successfully by Rind and Bramwell (1996) for various simulated collision and non-collision scenarios. In our first experiment, we compare the response of the continuous-time Rind model with the original to show that our implementation is also selective to collision over non-collision cases. A computer-generated 2D stimulus was simulated to approach, recede, and fly past a stationary sensor plane as detailed in Section 2.4.1.

In the first two cases, we varied the speed of a simulated bright object (object-background contrast was one) from 2.8 m/s to 14 m/s in steps of 2.8 m/s. The output of the LGMD unit in the continuous-time Rind model was recorded for both an approaching and a receding object. The response of the continuous-time Rind model and the original Rind model are shown in Figure 2.6. The output of the LGMD unit increases exponentially as the object approaches the insect, and peaks when the object occupies the entire visual field (see Figure 2.6a). Also, the fastest approaching object was seen to elicit the strongest and quickest response. The response for comparable speeds in both the continuous-time and the original Rind models match closely (see Figure 2.6a and b). The only difference is that the response of the continuous-time Rind model has a ‘bump’ right at the start of the approach which is missing in the original model. This represents the flash response of the LGMD neuron (Rind and Bramwell, 1996) and is visible for the receding object case in the original model (see Figure 2.6d). The flash response is dependent on the initialization of the filter buffers of the continuous-time Rind model and can be eliminated if the stimulus is kept static for some initial time before the approach. The period of time for which the object must be kept static is equal to the buffer length of the slowest filter used in the model. This is equivalent to initializing the buffers of all the filters being used in the model as described in Section 2.4.2. The response of the continuous-time model to receding objects was almost zero when simulated for the same set of speeds, except for the flash response at the start of the simulation. Notice that the flash response in the continuous-time Rind model is stronger than the flash response in the original Rind model (see Figure 2.6c and d). This may be due to our initializing the filter buffers to zero (black). Since the contrast of the object was one in this experiment, the simulated photoreceptors saw a bright flash (black to white transition), hence a larger flash response.

The response of the continuous-time Rind model was unchanged when the contrast of the object and the background were reversed (a black square on a white background). The response of the model to an accelerating object was stronger than the response to an object approaching at a fixed speed. This is also in accordance to what is known about the original Rind model: the faster the

edges of the object expand, the stronger is the output of the LGMD unit. In all these cases, the continuous-time Rind model was equivalent to the original Rind model.

The response of the continuous-time Rind model with rectification, and with motion inputs are roughly the same as the above model for the collision and non-collision cases discussed above. The STI model has also been shown to successfully detect collision over non-collision in various scenarios by Harrison (2005).

In the next experiment, we probe the model with real-world car videos as stimuli and compare the response characteristics with both the continuous-time Rind model with motion input and the STI model.

2.5.2 Experiment 2

In this experiment we used video clips as input that were recorded from a stable camera inside a car. The car videos were those used by Cuadri *et al.* (2005b) (detailed in Section 2.4.1). The videos were recorded at a rate of 25 frames per second. These videos show three scenarios: a car undergoing a collision, horizontal motion in front of a stationary car, and a car driving around without encountering any collisions. We created a reverse-collision video by replaying the collision video in reverse time to generate a scenario similar to that of a receding object in Experiment 1. The continuous-time Rind models and the STI model were simulated with these video clips as inputs and their responses are presented below.

Our experiments with the continuous-time Rind model revealed that it was not able to discriminate between the collision and non-collision videos. This is consistent with what other authors have reported about the original Rind model for the same videos (Cuadri *et al.*, 2005a; Stafford *et al.*, 2007b). Here, we compare the performance of the continuous-time Rind model with rectification, continuous-time Rind model with motion input, and the STI model for the same car videos.

The response of the continuous-time Rind model with rectification using these video clips are shown in Figure 2.7. All the buffers in the filters were initialized by running the simulation for the first six seconds using the first frame as input. For a collision scenario, the response of the model is shown by a dark line. This response weakly follows the approach of the obstacle and peaks before the collision. A much stronger peak response was recorded for the same movie played in reverse time (thin line). This strong response was due to the inability of the lateral inhibitory network to suppress sudden appearance of the edges of the object in the visual frame. The response to two horizontally translating cars (dashed line) was also about twice as strong as the collision case output. However, the peak response to a non-collision car movie (thick line) was about 50% smaller than to the collision response. This non-collision movie included rotational input (when the car turned around corners) as well as input from slowly moving objects (cars moving in the same direction but in different lanes).

This experiment reveals two major problems with this model: i) the reverse-collision peak response is larger than any other case by a factor of three, and ii) the horizontally translating cars were able to produce twice as strong a peak response than the collision case. To suppress the horizontal motion component in the model, we simulated the continuous-time Rind model with motion input which utilizes HR units to compute motion components in radially outward directions and feeds them to the subsequent stages of the continuous-time Rind model with rectification (for details see Section 2.4.2).

The output from the continuous-time Rind model with motion input is shown in Figure 2.8. The response to the collision video (dark line) was stronger compared to the horizontally translating cars (dashed line) and the non-collision video (thick line). However, the strongest peak response was still recorded for the reverse-collision case (thin line). The response was strong in the reverse-collision case because of sudden appearance of object edges in the visual frame. Leaving aside the

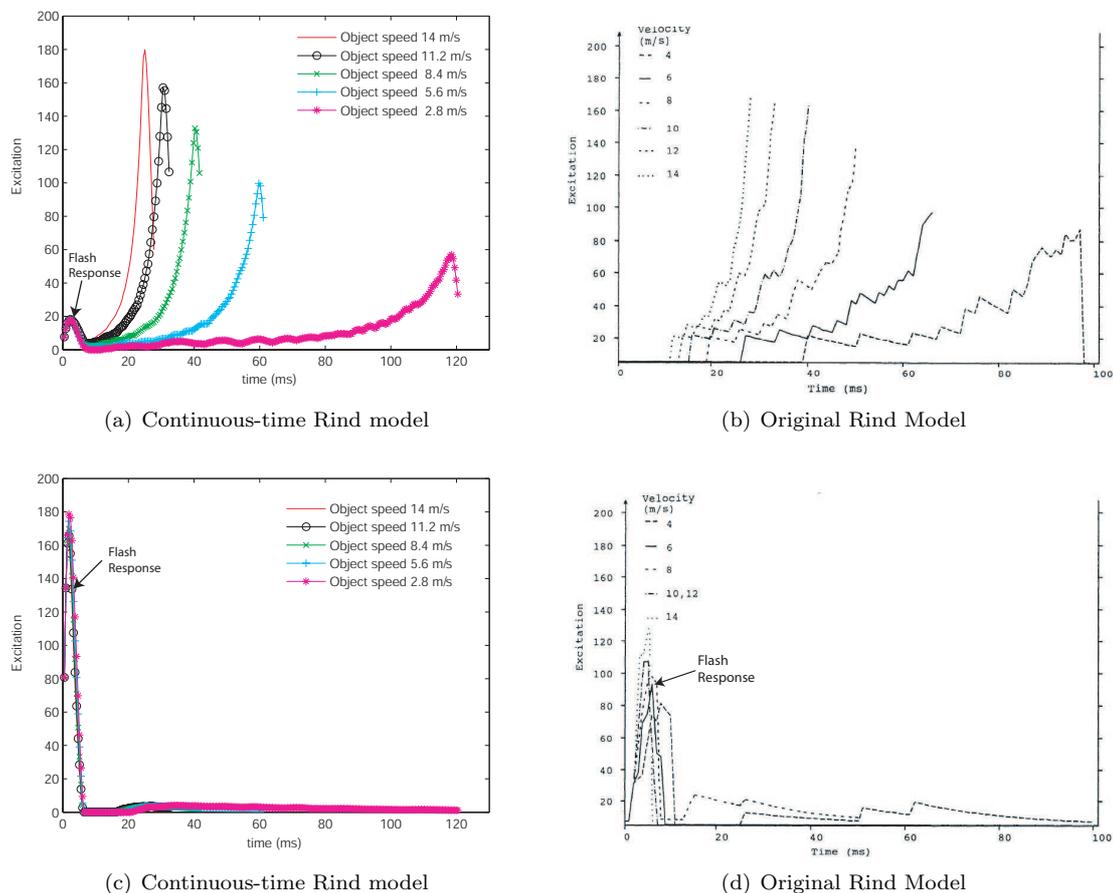


FIGURE 2.6. Comparison of the response from the continuous-time and original Rind model. (a) Response of the continuous-time Rind model to an approaching object. The speed of approach was varied from 14 m/s to 2.8 m/s. The peak was observed when the object occupied the entire visual field. The appearance of the object in the visual field causes a small peak (flash response) in the initial 8 ms. (b) Response of the LGMD unit in the original Rind model. Approach speed varied from 14 m/s to 4 m/s. The original model does not have a flash response for an approach scenario. (c) Response of the continuous-time Rind model to a receding object at same speeds as in (a). The sudden appearance of the object in the visual field elicits a flash response from the model. The response decays to zero after the flash response. (d) Response of the original Rind model to a receding object at same speeds as in (b). There is a spike in the excitation level due to the sudden appearance of the object in the visual field similar to the flash response in the continuous-time model.

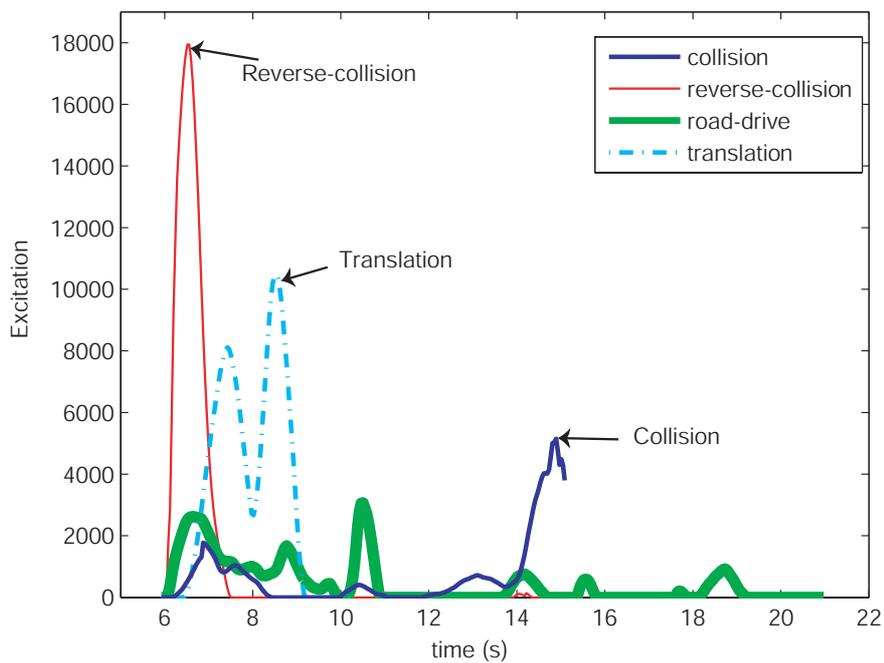


FIGURE 2.7. Response of the continuous-time Rind model with rectification to multiple car videos. The thin line represents the response of the system to a reverse-collision video. The dark line represents the collision case, while the dashed and the thick lines represent translating-cars video and non-collision road-drive video, respectively. The responses to the reverse-collision and translation videos are stronger than the response to the direct collision video due to the inability of the lateral inhibitory network to suppress fast translation of edges across the visual field.

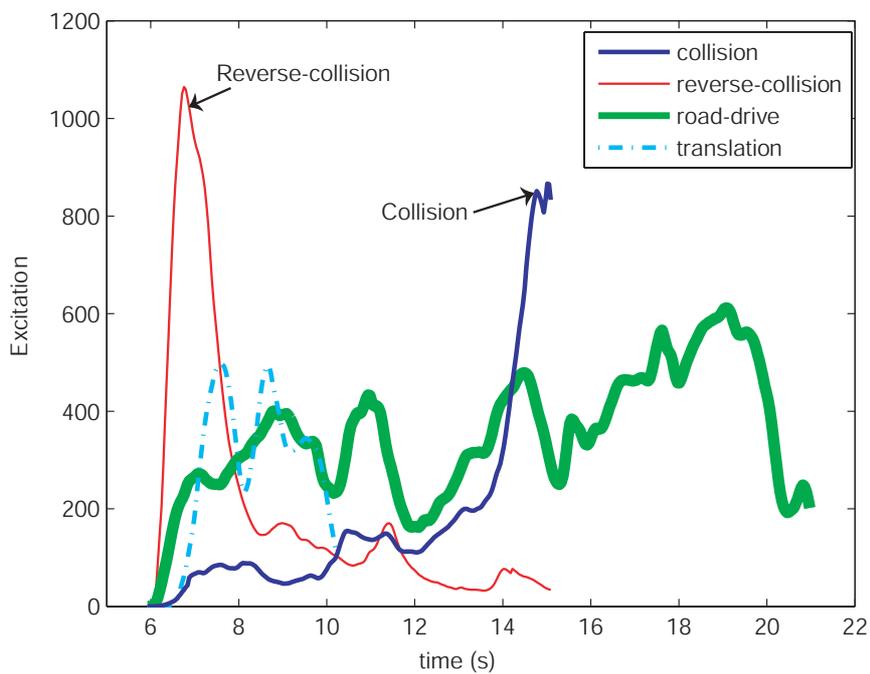


FIGURE 2.8. Response of the continuous-time Rind model with motion input to multiple car videos. The thin line represents the response of the system to a reverse-collision video. The dark line represents the collision case, while the dashed and the thick lines represent translating-cars video and non-collision road-drive video, respectively. The collision response is stronger than the response to the translation and road-drive videos. The response to the reverse-collision video is the strongest and shows that expansion is not sufficient to suppress large sudden motion of edges.

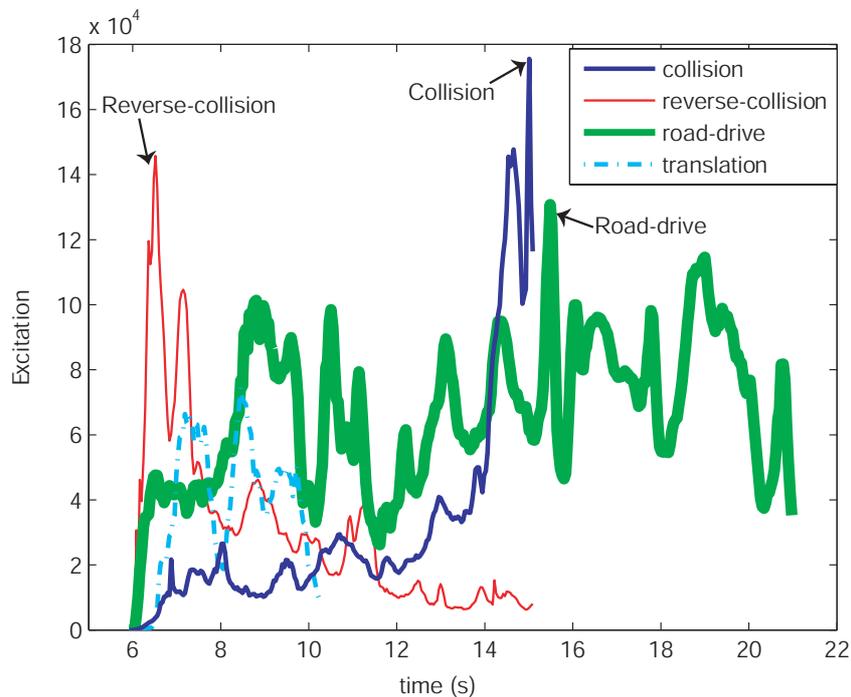


FIGURE 2.9. Response of the Spatio-temporal integration model to multiple car videos. The thin line represents the response of the system to a reverse-collision video. The dark line represents the collision case, while the dashed and the thick lines represent translating-cars video and non-collision road-drive video, respectively. The response to the collision video is the strongest. The response to the non-collision and reverse-collision videos are within 30% of the collision response.

reverse-collision case, this model performed notably better than the simple continuous-time Rind model.

Next we simulated the STI model for the same set of car videos. The response of the STI model to the different car videos is shown in Figure 2.9. Unlike the response of the continuous-time Rind model with motion input, here the response to the collision video was the strongest (dark line). The response to the reverse-collision video (thin line) spiked initially but the response was smaller than the peak response for the collision case. The responses to the horizontally translating cars (dashed line) and the non-collision video (thick line) were weaker and distinguishable from the collision case. This model was easily better than the other two models and a threshold could be set to separate the collision and the non-collision cases which is the goal of any successful collision avoidance model.

In the next section, we analyze the response characteristics of the three models simulated in these experiments and discuss the relative performance of these models.

2.6 Discussion

As seen from the simulation results with computer-generated stimuli, the continuous-time Rind model relies on the near-exponential increase in the size of an approaching object to trigger a collision avoidance response. If the object is not on a collision trajectory (for example, if it is translating across or receding from the observer) then the lateral inhibition layer suppresses the response of

the LGMD unit. This way, only a fast approaching object triggers a collision avoidance response. All other motion responses are suppressed. In a real-world scenario, however, rotation and/or fast translation of the observer also causes a large number of summation units (S) to activate, triggering a collision avoidance response. This false alarm is caused mainly because the Rind model operates by calculating local edge speed. It relies upon the edge information provided by the input layer (HPF units) and the delayed inhibition mechanism of LPF_1 units to predict an impending collision. Being non-directional, it makes no distinction between a rotation, a forward translation, or a sideward motion. This was discovered during many failed simulations using the continuous-time Rind model in our study. This shortcoming of the model has also been reported by other authors (Cuadri *et al.*, 2005a; Stafford *et al.*, 2007b). They have argued that for an insect, an approaching predator has a much larger size-to-speed ratio than any self-induced motion. The car videos that we used in our simulation study did not represent the large speed-to-size ratio of a predator. In the case of the car collision video, for instance, the expansion of the obstacle (a toy car) was not as rapid as a predator approaching its prey as can be seen by the time course of the simulation (Figure 2.7). This was also the reason why the continuous-time Rind model with rectification did not elicit a very strong response to the collision video.

In the case of the translation video, the horizontally translating cars moved across the camera at a very close distance (large edge extent) and at a fast speed. Due to the fast speed of the cars, the lateral inhibition units did not have enough time to suppress the activity of the LGMD unit. The response of the continuous-time Rind model with rectification to the translation video was much stronger than that to the collision video. The continuous-time Rind model with motion input, on the other hand, was able to suppress the translational motion across its view field as it was filtered out by the expansion filter (see Figure 2.5). Therefore, the peak response of the continuous-time Rind model with motion input to the translational video was smaller than its peak response to the collision video by a factor of two. The STI model is exactly similar to the continuous-time Rind model with motion input up to the expansion-only filter. Therefore, it is not surprising that the response of the STI model to the same video clip is also much smaller than its response to the collision video.

We also notice a strong response to the reverse-collision video for all the models. This was due to the manner in which the car collision was recorded. The camera was focused on the car and the car occupied the entire visual field at the start of this video clip. In the first one second of the movie, as the camera backed up, the outer edges of the car appeared in the frame for the first time. Since the output of the continuous-time Rind model is proportional to the edge extent visible in the field of view, the response increased during the first thirty frames. This was also the reason for a stronger response in the case of continuous-time Rind model with motion input and the STI model.

Based on the above comparisons, we can conclude that no threshold can be set for the continuous-time Rind model with rectification that will disambiguate between the collision and non-collision scenarios. The continuous-time Rind model with motion input is more attractive, and in our simulation study fails only for the case when the car video was played in reverse time and the obstacle was moving very close to the camera. The STI model performed the best among the three models. Since the continuous-time Rind model with motion input uses the STI model as a sub-unit, we may note that its additional processing units actually make it respond worse than the STI model. The lateral inhibition in the continuous-time Rind model with motion input suppresses the response of the S units even for the collision case. However, the lateral inhibitory network is too slow to respond to the sudden appearance of the edges in the case of reverse-collision video. This caused the response to the reverse-collision video for the continuous-time Rind model with motion input to be larger than the collision video. In the STI model, there is no lateral inhibition, and therefore the collision output is not suppressed in any manner and is the strongest among all videos.

2.7 Summary

We presented a simulation study of two novel continuous-time implementations of the Rind model and compared their response to the STI model. The STI model and the continuous-time Rind model with motion input performed better than the continuous-time Rind model with rectification for the horizontal translation video. This was because the continuous-time Rind model with motion input and the STI model have a preference for expansive motion. The input to the continuous-time Rind model with motion input is processed by an expansion filter, thereby reducing the response due to pure translation. The STI model is structurally the same as the continuous-time Rind model with motion input up to the expansion filter. Nevertheless, the sudden appearance of a lot of edges, as in the case of the reverse-collision movie, was still enough to trigger a large response from all the models. In the case of both the continuous-time Rind model with motion input and the continuous-time Rind model with rectification, the responses were larger than the responses to the collision movie. The response of the continuous-time Rind model with rectification was the smallest for the non-collision movie among all three models. The difference in the magnitude of the STI model's response to the collision, reverse-collision, and non-collision videos was within 30%. This may not be sufficient to determine a threshold for collision avoidance if the illumination and/or the speed of the car changes. Therefore, it can be argued that filtering the expansive motion alone may not be sufficient to generate a reliable avoidance response.

As discussed in Section 2.6, the lateral inhibition network in the continuous-time Rind model with motion input made it perform worse than the STI model. The response of this model was also worse than the continuous-time Rind model with rectification for the road-drive video with no collisions. The only place where the motion input based model was better than the rectification based model was the case of horizontally translating cars. In the next chapter, we use tracking to negate horizontal motion parallel to the observer. This makes the continuous-time Rind model with rectification more attractive than the continuous-time Rind model with motion input. We investigate the continuous-time Rind model with rectification and the STI model by means of mathematical analysis and show how tracking may improve their performance.

CHAPTER 3

THE EFFECT OF TRACKING ON COLLISION AVOIDANCE
PERFORMANCE

In this chapter, we explore in greater detail why the two collision avoidance models simulated in the previous chapter fail to reliably predict an imminent collision. The simulation results show that even when the models elicit a peak response to an approaching obstacle, the response magnitude by itself is not consistent for all collision or non-collision cases. In a threshold based model this poses a problem and raises an important question: *is it possible to relate the response magnitude of the collision avoidance models to the distance of the obstacle from the observer, and if so, how?*

We try to answer this question by developing a basic algebraic formulation of the problem where an obstacle on a collision course approaches a fixed viewer. The trajectory charted by an obstacle in the real world is three-dimensional (3D). However, the image that is processed by the insect visual system is only a two-dimensional (2D) projection of the real world. While the motion of the obstacle in the real world is described by three cartesian directions x , y , and z , the motion projected on the photoreceptors is possible only along two of the three directions (say, x and y). The motion along the third direction (z axis) is not explicitly resolvable by processing a sequence of two dimensional images (Duric *et al.*, 1999). The relative distance of the objects in a scene may be computed *directly* by using non-directional speed and motion parallax (Bruckstein *et al.*, 2005). We describe two such models for non-directional speed estimation in Chapter 5. In this chapter, we show how to *indirectly* relate the response of the collision avoidance models to the distance from the obstacle through tracking. This is inspired by the active vision approach used in computer vision (Bajcsy, 1985). We compare simulation results of both the continuous-time Rind model with rectification and the STI model with and without tracking to show how tracking improves the performance of the models. From here on for simplicity, we will refer to the continuous-time Rind model with rectification as the Rind model.

In the next few sections, we present a 1D and 2D analysis of the collision avoidance problem.

3.1 Analysis of a 1D object's approach

A common method used to analyze a collision scenario is to assume that a 1D object is translating towards an observer in a 2D world (Laurent and Gabbiani, 1998; Gabbiani *et al.*, 1999). The analysis below is based on such a 1D approach scenario.

Let us consider a rigid object of height h is approaching an observer E at a speed v (see Figure 3.1). Let the observer be able to compute the angle subtended by the object. At time t_1 , the object is at a distance d_1 from the observer and its angular subtense is θ_1 . At another time instant t_2 ($t_2 > t_1$), let the distance between the object and the observer be d_2 and the angle subtended by the object be θ_2 . Note that $0 \leq \theta_1, \theta_2 \leq \pi/2$. We can represent the distance from the observer as a function of the height of the object and its angular subtense.

$$d_1 = h \cot(\theta_1) \tag{3.1}$$

$$d_2 = h \cot(\theta_2) \tag{3.2}$$

The speed of approach v is then given as

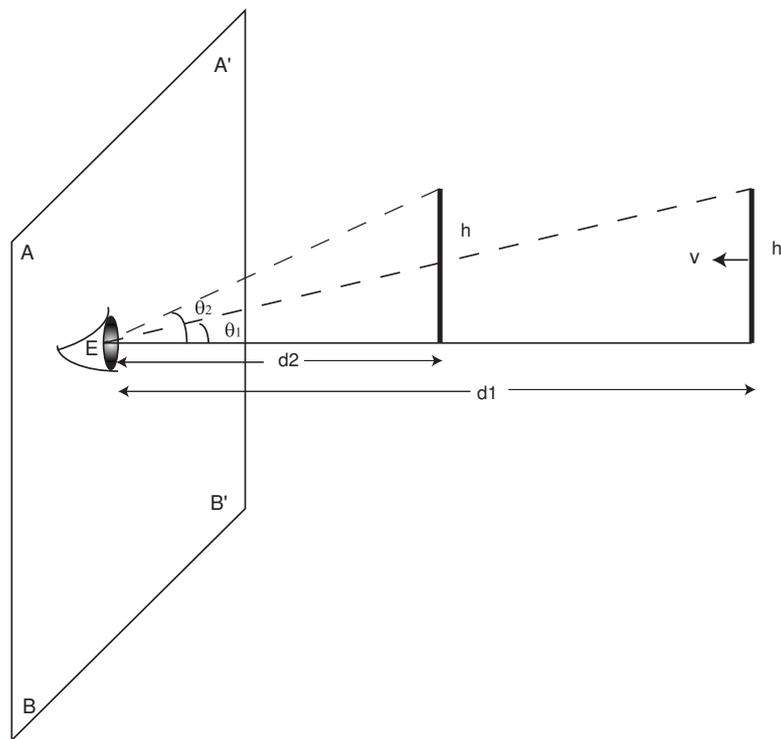


FIGURE 3.1. A 1D object's approach towards a fixed observer. The object of height h subtends angles θ_1 and θ_2 at two different times instants t_1 and t_2 during its approach towards the observer.

$$v = \frac{d_2 - d_1}{t_2 - t_1} \quad (3.3)$$

Replacing d_1 and d_2 with Equations 3.1 and 3.2, we get

$$v = h \cdot \frac{\cot(\theta_2) - \cot(\theta_1)}{t_2 - t_1} \quad (3.4)$$

In order to compute the time-to-collision (TTC) we simply substitute $\cot(\theta_2)$ with $\cot(\pi/2)$ (as θ_2 becomes $\pi/2$ when the object reaches the observer) and t_2 with t_c , the time of contact. From Equation 3.4 we get

$$t_c - t_1 = \frac{h}{v} (\cot(\pi/2) - \cot(\theta_1)) \quad (3.5)$$

Or simply,

$$t_c - t_1 = \frac{h}{v} \cot(\theta_1) \quad (3.6)$$

The TTC is, thus, dependent upon the measurable θ_1 and the unknowns h (size of the object) and v (the speed of the object). Though h and v are independently unknown, we can compute the ratio of the two from Equation 3.4 as:

$$\frac{h}{v} = \frac{t_2 - t_1}{\cot(\theta_2) - \cot(\theta_1)} \quad (3.7)$$

Therefore, in an ideal 1D approach scenario, it is possible to predict the time to collision by measuring the angular subtense of the object. This computation, however, is dependent on two implicit assumptions:

1. That the approaching object is in the direct collision path, and
2. That the speed of approach is constant at all times.

If either of the above assumptions is not met, the TTC estimate is no longer valid. For example, if an object is **not** moving directly towards an observer, then the TTC estimate only reveals the time it will take the object to reach an infinite sensor plane ($AA'BB'$ in Figure 3.1) passing through the observer. Instead of computing time to collision, in this scenario the TTC is the time left before the object passes by the observer. This is not particularly useful, especially when an observer is moving with respect to their surroundings. For instance, an observer moving through a tunnel with textured walls will not be able to predict whether the looming texture on the wall is harmlessly passing by or is going to collide with it. The η -function based model uses a similar analysis for computing both the η and TTC parameters. For an off-centered approach, the η -function based model may not predict a collision reliably. Additionally, we have not discussed how the angular subtense of an object would be determined from the intensity map incident on the photoreceptors. As discussed in the previous chapter, to obtain the angular subtense of an object, the visual system would have to segment the visual scene – a complicated processing step.

A 1D approach scenario in a 2D world is unrealistic and at best, only helps to provide a basic understanding of the problem. Most biological and artificial systems have a 2D array of sensors on which the 3D world gets projected. In the next section, we consider this problem of motion in a 3D world projected on a 2D array of sensors.

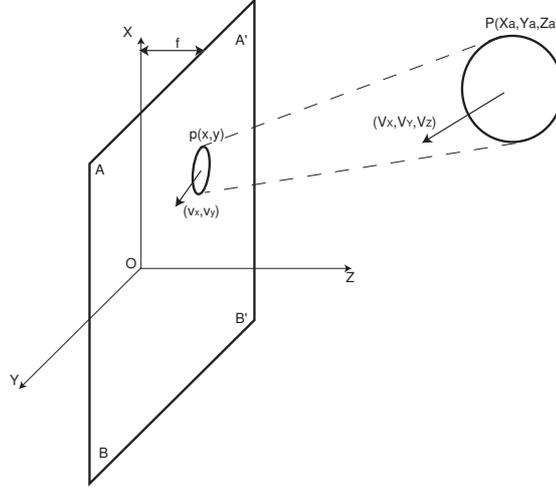


FIGURE 3.2. 2D projection of a 3D object. The point P in the real environment is projected as p on the sensor array.

3.2 Motion under 2D perspective projection

In a biological system, the visual input is usually modeled as a 2D projection of the real world onto an array of photoreceptors. The intensity map is then processed by successive stages of the visual system to interpret the scene (Rind and Bramwell, 1996; Franz and Krapp, 2000). The same is also true for most optical-flow/motion-flow based computer vision algorithms (Horn, 1986; Fermuller and Aloimonos, 1992). In this section, we present a mathematical framework to analyze how a model that only has access to the intensity information extracts parameters of object motion like speed and distance. As shown in Figure 3.2, the sensors are arranged in a 2D plane $AA'BB'$. Under perspective projection (Foley *et al.*, 1995) on a plane at a distance f from the origin, the motion at a point $P(X_a, Y_a, Z_a)$ with translational speed $S(V_X, V_Y, V_Z)$ is represented by the point $p(x, y)$ with speed $s(v_x, v_y)$, where

$$x = \frac{X_a f}{Z_a} \quad (3.8)$$

$$y = \frac{Y_a f}{Z_a} \quad (3.9)$$

Without loss of generality, we can choose $f = 1$ as the focal length of our imaging system. Then, the velocity at point $p(x, y)$ is computed by differentiating x and y and substituting X_a and Y_a according to Equations 3.8 and 3.9:

$$v_x = \frac{dx}{dt} = \frac{d}{dt} \left(\frac{X_a}{Z_a} \right) = \frac{V_X}{Z_a} - \frac{x V_Z}{Z_a} \quad (3.10)$$

$$v_y = \frac{dy}{dt} = \frac{d}{dt} \left(\frac{Y_a}{Z_a} \right) = \frac{V_Y}{Z_a} - \frac{y V_Z}{Z_a} \quad (3.11)$$

Thus, we find that the speed of the projected point p on the sensor array is a combination of the speed in the respective cartesian (x or y) direction and the distance of the object normal to the sensor plane (z direction). These equations represent an underdetermined system with an infinite number of possible solutions. This becomes clear by looking at the geometrical interpretation of Equations 3.10 and 3.11. These equations represent planes in the (V_X, V_Y, V_Z) coordinate system (see Figure 3.3). The solution to the system of equations is represented by the intersection of the two planes – a line L . *Thus, motion along the x and y directions (parallel to the sensor plane) is indistinguishable from the motion along the z direction (approach).*

If it were possible to remove the lateral motion of the object, its motion towards the observer would be exposed. However, the discussed scenario of a fixed viewing plane has an inherent limitation in the way information about the motion of the real world object is represented at the very first stage as discussed in the previous paragraph. Also, this representation is dependent on how accurately we can determine the local velocity of the point p .

Optical flow, which is defined as the local velocity field of point objects in a scene, has been argued to be a good mechanism to determine the motion of rigid body objects (Horn, 1986; Fermuller and Aloimonos, 1992). However, the computation of the optic flow is by itself an ill-posed problem and simplifying assumptions, such as smoothness of the intensity pattern, are required to solve the mathematical equations (Horn and Schunck, 1981). Spatio-temporal frequency based algorithms, on the other hand, are stable correlation-based methods that can be used to compute local motion at each point in a scene (Verri and Poggio, 1989; Lindemann *et al.*, 2005). It has been shown that spatio-temporal frequency based methods such as the HR model may also be used for estimating the qualitative properties of motion from the real world, such as the focus of expansion, or discontinuities due to relative motion of an object with respect to its background (Verri and Poggio, 1989). In some cases, including collision detection, these qualitative properties may be used instead of quantitative optical flow to determine the trajectory of an approaching obstacle. For these reasons, correlation-based models like the STI model are superior to computer vision based algorithms that mostly rely on exact optical flow calculation (Horn, 1986; Aloimonos *et al.*, 1988; Fermuller and Aloimonos, 1992).

3.3 Collision avoidance models - an analysis

The use of correlation-based models in place of optical flow based algorithms is advantageous only to the extent that it presents a mathematically stable and computationally simple alternative (Reichardt *et al.*, 1988; Verri and Poggio, 1989). However, it does not address the issue that the local speed of a point on the sensor plane may not be used to disambiguate between the infinitely many solutions for the corresponding motion in the real world. With this background, we present an analysis of the Rind and STI models to reveal their specific limitations in the hope of finding solutions to address these shortcomings.

3.3.1 The Rind model

As has been discussed in some detail in the previous chapter, the Rind model relies on a lateral inhibitory network to reject the non-collision scenarios from the collision ones. The basic assumption of the model is that an object on a collision course expands more rapidly than an object translating parallel to the observer, and hence, only the object on a direct collision path will be able to overcome the lateral inhibition from the neighboring sensor units. However, the simulation results from the previous chapter show that unambiguous determination of a threshold for such a collision avoidance model is not trivial. Let us consider the simple case of an object moving with a velocity (V_X, V_Y, V_Z) in the real world with the sensor plane of the Rind model oriented along the x - y plane. Let us choose the approach speed (V_Z) of the object to be zero. In this case of non-collision, the output of the Rind model may still increase due to any of the following reasons, illustrated in Figure 3.4:

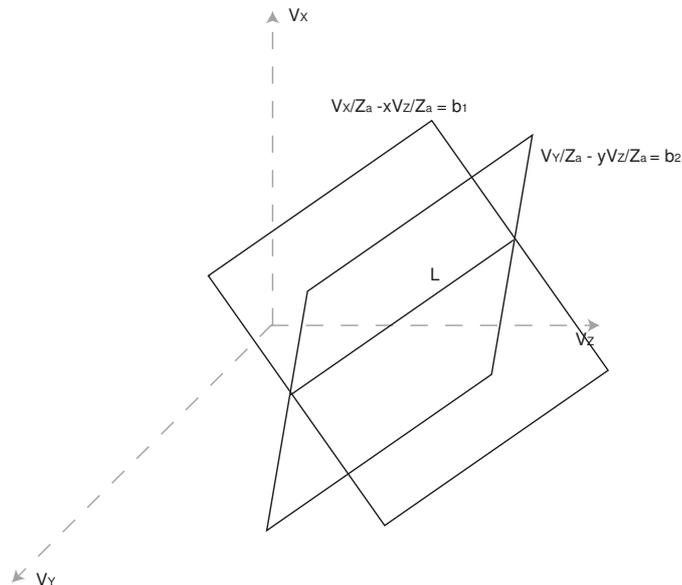


FIGURE 3.3. Geometrical interpretation of an underdetermined system of equations. The planes parallel to the x - z and the y - z surfaces intersect each other at a line L .

- the contrast of the object may be changing due to its non-uniform surface reflectance and/or due to non-uniform illumination,
- the object may be entering the visual field of the observer, and hence an increasing number of motion units may be getting activated,
- the translational speed of the object may be faster than the delay in the inhibition network.

In the first scenario, the increase in response due to non-uniform surface reflectance or illumination is caused by the absence of a contrast saturation scheme in the Rind model. Evidence of contrast saturation in the insect visual system has been found and modeled (Harris *et al.*, 2000; Rivera-Alvidrez and Higgins, 2005). This problem might be solved by simply incorporating a similar contrast saturation unit in the Rind model.

The inability of the Rind model to reject motion parallel to its sensor plane is the main reason behind the increase of the model's response in the last two cases. Another problem that ties into this set is the case when an object is approaching from a wide angle with respect to the sensor plane. In such a case, the object may not be in the view field of the observer initially, and based on the angle of approach it may not award sufficient time to the observer to elicit an avoidance response. All the above-listed problems may have a common solution *if the observer may somehow keep the object in the center of the visual field*. We will discuss this further in Section 3.4.

In the next section, we examine the STI model.

3.3.2 The STI model

The STI model also has its own set of limitations. The model computes the expansion in a visual scene to determine whether or not a collision is imminent. As shown in Section 3.2, the motion computed from the visual image of a scene does not disambiguate between translation parallel to the observer and approach. This becomes apparent when we observe how the expansion output is

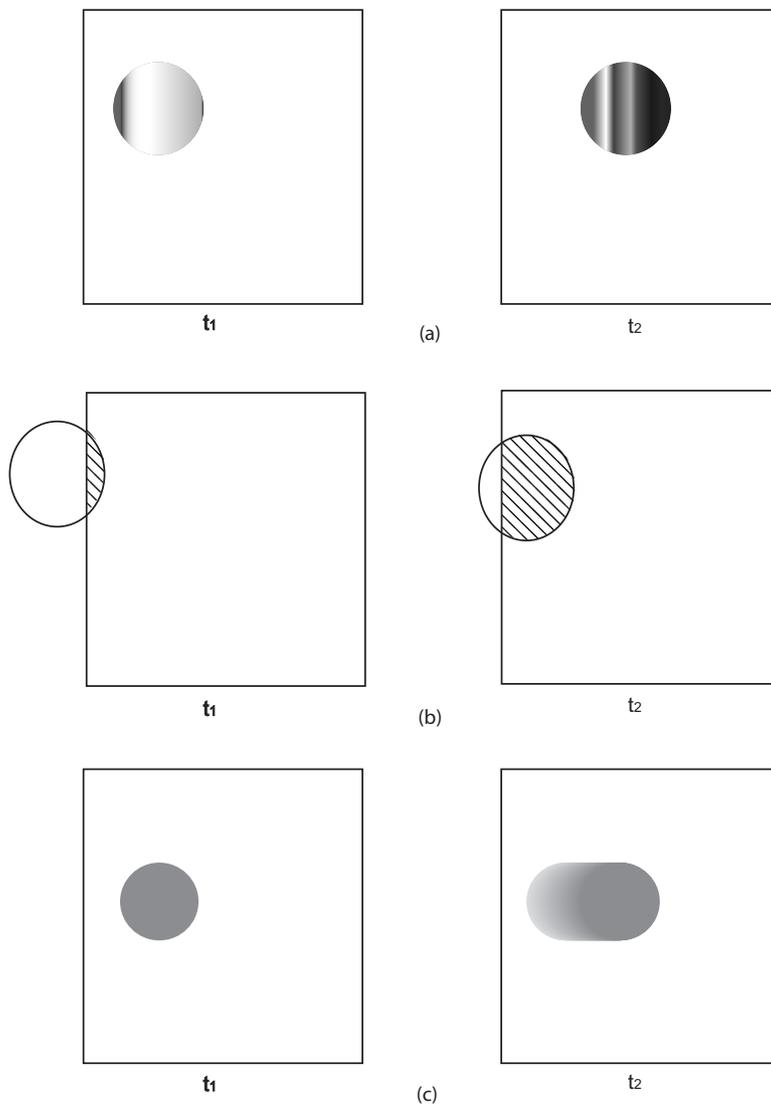


FIGURE 3.4. Situations which limit the collision avoidance performance of the Rind model. Each pair of boxes represents the 2D image of the real world at time instants t_1 and t_2 ($t_2 > t_1$). (a) Non-uniform surface reflectance or illumination may lead to an increased activation of the motion units. (b) The edge extent of an object within the visual field of an observer may change due to the object's entry into the view field. (c) A fast moving object may activate a larger set of motion units due to slow lateral inhibition.

computed on the input visual image. The 2D sensor array extracts motion information in both the cartesian x and y directions before combining them to calculate the net motion at a given point. The net expansion is computed by combining the outward motion in each quadrant as described in Equation 2.6.

Figure 3.5 shows four common scenarios experienced by a navigating observer: (a) expansive motion field in the case of a direct collision, (b) rotational motion field while rotating about a fixed axis, (c) off-center expansion in the case of an oblique collision, and (d) self-induced motion field while driving forward in a textured tunnel. We can compute the net expansion for each case normalized to the value of the direct collision scenario by utilizing Equation 2.6. Thus, the normalized net expansion for the direct collision case is 1. For rotational motion, this may be computed to be 0.65. For the off-center expansion shown in Figure 3.5c, the net expansion is only 0.0952! In the case of a forward drive in a textured tunnel, the net expansion is 0.894.

Based on the computation of net expansion in four commonly encountered scenarios we find that net expansion alone may not be used for calculating a threshold for collision. In the above scenarios, the response of the model to an oblique collision was much smaller than the case of a benign forward drive through a textured tunnel. The response due to pure rotation was also about half as strong as the direct collision case. If the rotational speed is more than the direct collision speed, then the response due to a rotation may exceed the response due to a collision.

Thus, we see that both the Rind and the STI models fail to reliably detect an impending collision and may lead to false alarms or may fail to report an imminent collision. The major reason behind the failure in the Rind model is its inability to reject horizontal motion, and the STI model is the manner in which it computes expansion. Both the problems may have a common solution based on tracking the approaching object in the visual field. We discuss such a solution in the next section.

3.4 Tracking and collision avoidance

Many researchers in the field of computer vision have argued how an ‘active’ observer may be able to tackle some of the problems we have discussed in the previous section (Bajcsy, 1985; Aloimonos *et al.*, 1988; Fermuller and Aloimonos, 1992). An active observer is an observer that is free to move its ‘gaze’ in order to track some feature or object in its visual field. Figure 3.6 shows an active observer that has two degrees of freedom about its vertical and horizontal axes. Such an observer is able to rotate its sensor plane such that it may keep an object directly in front of it. As shown in Figure 3.6, this may be accomplished by keeping the line passing through the center of the sensor plane (P) and the center of mass of the object (P') normal to the sensor plane.

Let us examine how this affects the computation in the case of our two representative collision avoidance models. To answer this, let us consider the case of an object moving in the real world with speed (V_X, V_Y, V_Z) . Let the speed of approach be zero ($V_Z = 0$). In this case the object is moving in a two dimensional plane parallel to the viewing plane. Let us also assume that the observer is equipped with an algorithm that can exactly track the object and is able to compensate for the motion of the object almost instantly. In this scenario, as long as the observer is able to reject its self-motion induced due to rotation of its sensor plane, the only motion it experiences is due to the change in the viewing angle of the object. This component is usually much smaller than actual motion in any direction. Therefore, the response of the collision avoidance models for horizontal motion will be greatly diminished.

Next, let us consider the case when $V_Z \neq 0$. Since the observer is able to compensate for the motion in the $x - y$ plane, the only motion registered by the sensor array is due to the motion in the z plane and due to a change in the viewing angle. If we neglect the much smaller motion component due to the change in the viewing angle as compared to the expansive motion in the z direction, the problem of 3D motion gets converted into a 1D motion problem. This is the same scenario as was discussed in Section 3.1 where the object is always maintained in the center of the visual field. The

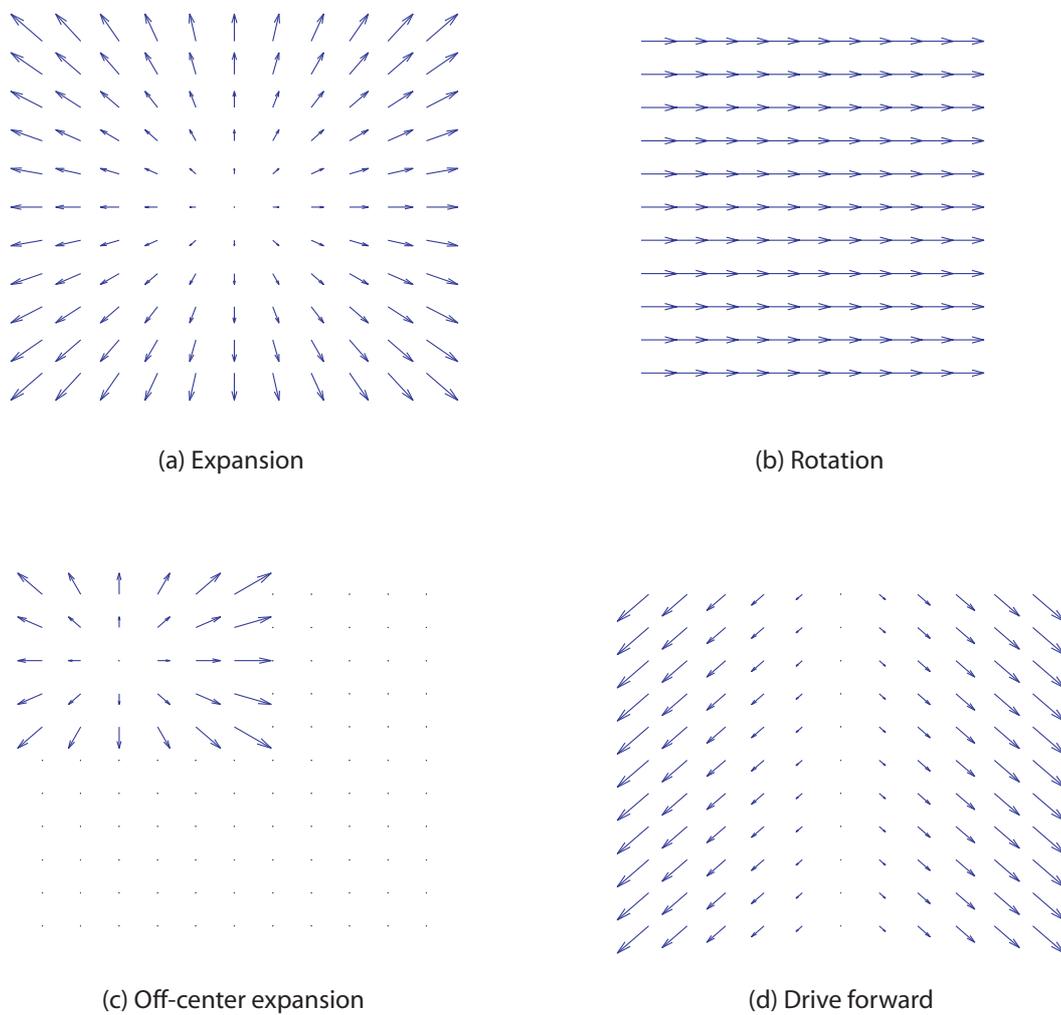


FIGURE 3.5. Limitations of the Spatial and Temporal Integration model. (a) Expansive motion of an object is computed by measuring the speed of the edges along x and y directions and combining them via vector summation. (b) A clockwise rotation of the observer about its axis generates horizontal motion which is consistent with the expansion motion in the right-half of the view field. (c) An off-center approach is not always consistent with the expansive motion output. (d) Outward motion of a pattern may be consistent with the expansive motion while the observer is moving in a tunnel with vertical stripes.

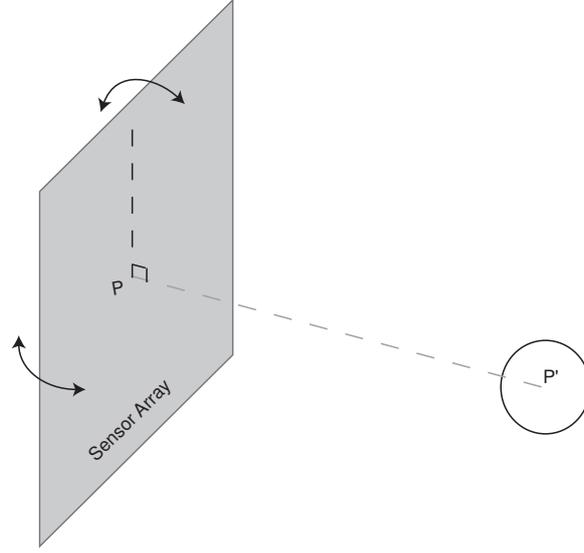


FIGURE 3.6. Active Observer with two degrees of freedom, shown by the arrows along the horizontal and vertical axis. The visual sensor may be rotated about either axis to actively track the object in its visual field.

tracking of the object makes $V_X = V_Y = 0$. Thus, the equation for motion of a projected point p on the sensor plane, as shown in Figure 3.2, is now simplified to:

$$v_x = \frac{dx}{dt} = \frac{V_X}{Z_a} - \frac{xV_Z}{Z_a} = -\frac{xV_Z}{Z_a} \quad (3.12)$$

$$v_y = \frac{dy}{dt} = \frac{V_Y}{Z_a} - \frac{yV_Z}{Z_a} = -\frac{yV_Z}{Z_a} \quad (3.13)$$

From these equations, it is possible to exactly compute both the speed of approach V_Z and the distance Z_a of the object from the sensor plane provided we can estimate the correct v_x and v_y values at the sensor plane location (x,y) . Thus, tracking an object in the visual field may solve most of the problems that we discussed about the Rind and STI models.

Several researchers have shown that it is possible to navigate without collision by using optical flow with visual tracking for a binocular observer (Bandopadhyay and Ballard, 1991; Aloimonos *et al.*, 1988). Fermuller and Aloimonos (1992) have developed a collision avoidance algorithm for an active monocular observer using normal flow (optical flow normal to the boundaries of an object). However, for this algorithm to work the normal flow must be exactly computed at each instant. As was discussed in Section 3.2, this is extremely hard for most real scenarios. Our contribution in this work is developing a framework for biological collision avoidance models using a spatio-temporal frequency based approach which does not suffer from the above problem.

In the next section we describe a simulation environment to compare the performance of the Rind and STI models with and without tracking capability.

3.5 Methods

We simulated the Rind and STI models, with and without a tracking algorithm, using Matlab version R2007a (The Mathworks, Natick, MA). The simulation environment was designed to test

the response of the models for collision and non-collision scenarios. In all the cases, a single two-dimensional object oriented parallel to the x - y cartesian plane moved inside a three-dimensional world. The size of the visual arena was scaled to match the scaling used in the simulations performed in Chapter 2. The size of the arena was calculated to be of size $4480 \times 4480 \times 5600 \text{ cm}^3$. A 40×40 array of sensor units implementing either the Rind or the STI model was used to compute a response of the moving object. Using the same scale as above, the size of the object was chosen to be $2.24 \text{ m} \times 2.24 \text{ m}$ such that it was equal to the extent of the sensor array. The implementation of the STI and Rind model were the same as described in Section 2.4.2.

For simulations of the Rind and STI models with tracking, the sensor array was kept pointing towards the approaching object if the object was in front of it. The sensor plane was allowed to rotate only $\pm 45^\circ$ about either axis. The orientation of the sensor array was determined by computing the centroid of the intensity map of the visual field using a feed-forward linear system. The tracking algorithm computed the centroid (x_c, y_c) using the following formula:

$$x_c = \frac{\sum_i x_i \cdot \sum_j E(x_i, y_j)}{\sum_i \sum_j E(x_i, y_j)} \quad (3.14)$$

$$y_c = \frac{\sum_i y_i \cdot \sum_j E(x_i, y_j)}{\sum_i \sum_j E(x_i, y_j)} \quad (3.15)$$

where $E(x_i, y_j)$ is the intensity of a point at image location (x_i, y_j) , i and j span the x and y dimensions of the sensor plane, and the midpoints of the sensor plane are taken to be the origin.

Tracking was performed by adjusting the azimuth and elevation angles of the sensor plane based on the centroid location at each timestep by using the following formula:

$$\theta_{az}(t) = \theta_{az}(t-1) + g_{az} \times x_c \quad (3.16)$$

$$\theta_{el}(t) = \theta_{el}(t-1) - g_{el} \times y_c \quad (3.17)$$

where the time instant t refers to the current frame and $(t-1)$ to the last frame, θ_{az} and θ_{el} are azimuth and elevation angles respectively, and g_{az} and g_{el} are gains for the respective cases.

Due to the simplicity of the visual environment, this algorithm computed an exact location of the target as long as the object remained in the visual field of the observer. We terminated the simulation whenever the object went outside the visual field of the observer, either because the sensor plane had to turn more than $\pm 45^\circ$ or if the object hit one of the walls in the 3D arena. The projected angular size of the object on the sensor plane was updated at each timestep such that its size grew inversely with its distance from the sensor array.

At the start of a simulation run, a simulated object was assigned a starting (x, y, z) position. The z position of the object specified the distance of the object from the viewer along a direction normal to the sensor plane. The x and y positions were $\pm 5.6 \text{ m}$ from the center of the visual field. The z positions were between 44.8 m and 56 m from the observer. The speed of the object along all three cartesian axes was then generated from a uniform random distribution. The range of speeds was chosen from three different sets for the x and the y axes. The first set was chosen between $\pm 1.68 \text{ m/s}$, the second between $\pm 2.24 \text{ m/s}$, and third between $\pm 3.36 \text{ m/s}$. The speed along the z axis varied from -2 to -20 m/s . The negative sign of the z speed denotes that the object always approached the viewing plane. A total of 1500 simulations were performed for each model. Due to the random generation of the speed, most trajectories resulted in non-collision scenarios. The object was considered close to an imminent collision at a distance less than 5 m from the observer (about twice the size of the object). The peak response (R_{peak}) of the models was recorded along with the distance (d_p) at which the response peaked and the minimum distance (d_m) an object reached to the observer. We binned this data for every 1.12 m from the observer and computed its mean (\bar{R}_{peak})

and standard deviation (σ_R). The \bar{R}_{peak} and σ_R were plotted versus both d_p and d_m . The data from the models with tracking capability were fit with the following equation for all but one case:

$$\bar{R}_{peak} = \frac{K_1}{d - d_0} \quad (3.18)$$

where d is either d_p or d_m , and K_1 and d_0 are fitting constants.

For the Rind model with tracking, \bar{R}_{peak} versus d_p curve was linear and was fitted with the equation of a line:

$$\bar{R}_{peak} = m \cdot d_p + c \quad (3.19)$$

where m is the slope and c is a fitting constant.

In Section 3.6.5, we present a case of noisy tracking to examine how sensitive the performance of the models with tracking capability are to noise in centroid computation. This was done by adding a Gaussian-distributed noise signal (mean = 0, variance = 8) to the centroid output of the tracking algorithm. We must note that the size of the sensor array is 40×40 and a variance of 8 is 20% of its size. Therefore, these simulations represent a very unreliable tracking algorithm. The simulations were performed with x and y speeds randomly chosen from a uniform distribution with speed between ± 2.24 m/s. The z speed varied from -2.68 to -13.44 m/s and a total of 500 simulation runs were performed for both the Rind and STI models with tracking.

3.6 Simulation results

In our investigations, we wanted to observe the peak response of the two models and the corresponding distance between the object and the observer. The aim was to analyze whether or not a thresholding scheme may be employed with a collision detection algorithm to judge an imminent collision.

3.6.1 Rind model without tracking

In this set of simulations a single 2D object approached the viewing plane from different starting positions within the virtual arena as described in the previous section. Figure 3.7a shows the mean and standard deviation of the peak response (\bar{R}_{peak} and σ_R) of the Rind model versus minimum distance from the observer (d_m). The plot was truncated to show data only nearer than 25 m from the observer. For distances greater than 25 m, the peak response was close to zero. The variance (σ_R^2) for distances smaller than 7 m from the observer are 72.5% to 113% of the mean \bar{R}_{peak} . These large variances suggest that the model often responded more weakly to a closer object than it did to a farther one. Another way of looking at the data is by plotting the \bar{R}_{peak} versus the distance at which that maximum response occurred (d_p). Figure 3.7b shows this data. We again note that for distances up to 12.5 m from the observer, the variance is sometimes larger than the \bar{R}_{peak} . For the case of an imminent collision (distance from the observer < 5 m, shown by a dashed vertical line in the figure), the variance of the model is 1.87 times its \bar{R}_{peak} and overlaps with the peak response values at distances greater than 10 m. The large variance of the peak response makes the determination of a threshold for collision impossible. This illustrates the limitation of the Rind model.

3.6.2 STI model without tracking

The STI model was simulated utilizing the same arena. Figure 3.8a show the \bar{R}_{peak} and σ_R of the STI model versus the minimum distance from the observer. Figure 3.8b show the \bar{R}_{peak} and σ_R versus the distance at which that response was elicited. The plots were truncated to show data only nearer than 25 m from the observer. As before, for distances greater than 25 m, the peak response was close to zero. Similar to the Rind model, we find that the \bar{R}_{peak} of the STI model has large

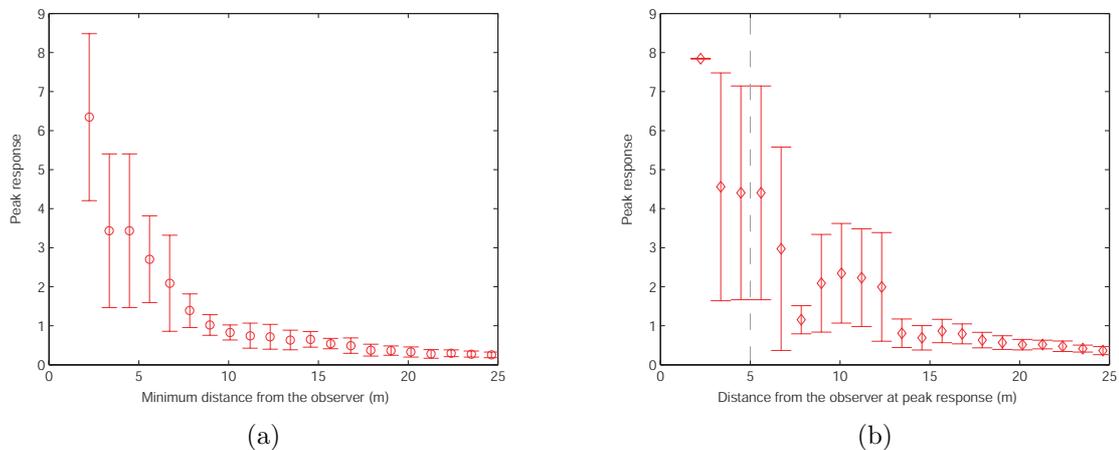


FIGURE 3.7. Mean peak response \bar{R}_{peak} (denoted by circles) and standard deviation σ_R (denoted by error bars) of the Rind model versus the distance from the observer. (a) \bar{R}_{peak} of the Rind model versus the minimum distance from the observer the object ever reached. (b) \bar{R}_{peak} of the Rind model versus the distance from the observer at which the peak response was attained. The dashed vertical line denotes the distance at which collision is imminent.

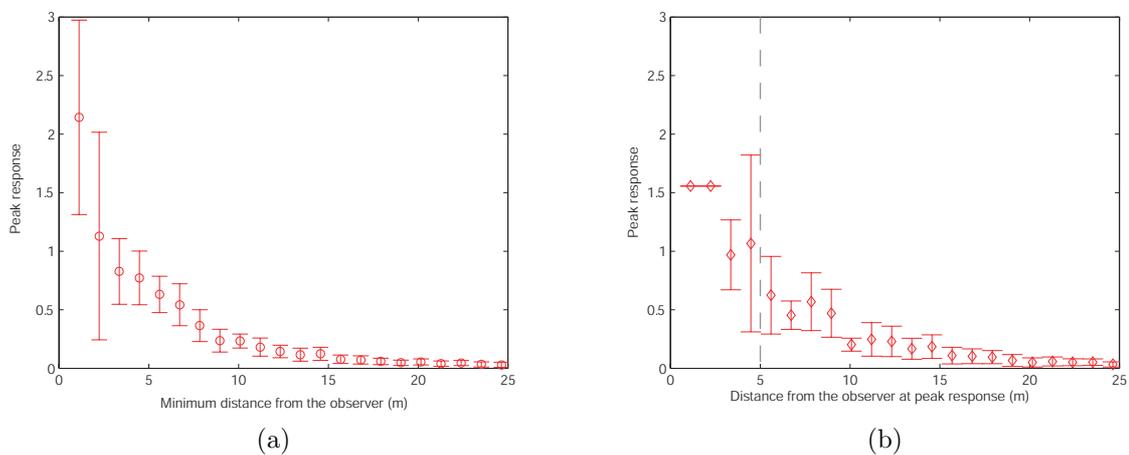


FIGURE 3.8. Mean peak response \bar{R}_{peak} (denoted by circles) and standard deviation σ_R (denoted by error bars) of the STI model versus the distance from the observer. (a) \bar{R}_{peak} of the STI model versus the minimum distance from the observer the object ever reached. (b) \bar{R}_{peak} of the STI model versus the distance from the observer at which the peak response was attained. The dashed vertical line denotes the distance at which collision is imminent.

variance. The first two data points in Figure 3.8b do not show any variance because they represent two individual cases and therefore, we do not include them in our analysis. For the case of an imminent collision (distance from the observer < 5 m, shown by a dashed vertical line in the figure), the σ_R is as large as 53% to 71% of the \bar{R}_{peak} value. The variance of \bar{R}_{peak} for these distances can be seen to overlap with the variance at distances larger than 10 m. This re-emphasizes the point that the threshold determination for the STI model for all possible collision scenarios is impossible.

Thus, we see that as expected from our analysis in Sections 3.1 and 3.2, both the models have severe limitations that must be addressed before they can be implemented as a threshold based collision avoidance system. We test the models with tracking capability in the next two sections.

3.6.3 Rind model with tracking

We simulated the Rind model equipped with a tracking algorithm as described in Section 3.5. The simulation sets were the same as those used for testing the Rind model without tracking. The \bar{R}_{peak} and σ_R for the Rind model with tracking versus the minimum distance from the object are shown in Figure 3.9a. The \bar{R}_{peak} values reach their maximum at a distance of 4.48 m from the observer. We have used Equation 3.18 to fit the data up to the peak with fitting parameters $K_1 = 8.45$ and $d_0 = 5.72$ m. To analyze why we see a maximum in the \bar{R}_{peak} values, we plotted \bar{R}_{peak} versus the distance at which the response was recorded as shown in Figure 3.9b. We find that the \bar{R}_{peak} increases linearly as a function of distance. A linear fit as described by Equation 3.19 with fitting parameters $m = 0.5945$ and $c = 11$ m is also plotted in the figure (dashed line). This plot does not show a roll-off which means that objects that came close may have been the ones that were flying past the observer not threatening a collision. The tracking algorithm allows the rotation of the sensor plane up to an angle of $\pm 45^\circ$ beyond which the simulation is terminated and the approaching object is deemed not a threat. These cases are responsible for the low values of \bar{R}_{peak} at distances less than 5 m in Figure 3.9a. However, the linear trend of \bar{R}_{peak} with large variance shows that even though the model performs much better than the one without tracking, the determination of the threshold is not trivial. The threshold may be set at an \bar{R}_{peak} value of 6. However, this may raise false alarms for some objects as far back as 8.96 m. We must note that we simulated scenarios that varied in approach speeds by almost one order of magnitude (2 m/s to 20 m/s) at random approach angles, and even though the response of the Rind model with tracking is far from ideal, it is still usable.

3.6.4 STI model with tracking

In this set of simulations, we simulated an STI model with tracking. The simulation parameters were the same as the ones used in simulating the STI model without tracking. The plot of \bar{R}_{peak} and σ_R for the STI model with tracking versus the minimum distance from the object is shown in Figure 3.10a. The \bar{R}_{peak} almost follows an inverse relation as seen in Equation 3.18 with fitting parameters $K_1 = 3.2$ and $d_0 = 1.24$ m. In Figure 3.10b, the \bar{R}_{peak} is plotted with the distance at which the response occurred. The data almost exactly follows the above mentioned equation with fitting parameters $K_1 = 2.5$ and $d_0 = 3.584$ m as shown by the dashed trend line in the figure. We find that even though the variance for the maximum \bar{R}_{peak} is large (distance < 5 m), it does not overlap with the range of values for distances outside that range. Therefore, the STI model with tracking may be used as a reliable threshold based collision avoidance system and the improvement with respect to the STI model without tracking is dramatic. Figure 3.10b shows that if the threshold is set at 1.6, all objects that pose a threat of imminent collision (distance < 5 m) will be detected. The better performance of the STI model with tracking with respect to the Rind model with tracking is due to its explicit computation of expansive motion. Once the tracking model neutralizes any motion in the x - y plane, the expansive motion computed on the projected image exactly estimates the motion towards the viewing plane as discussed in Section 3.4.

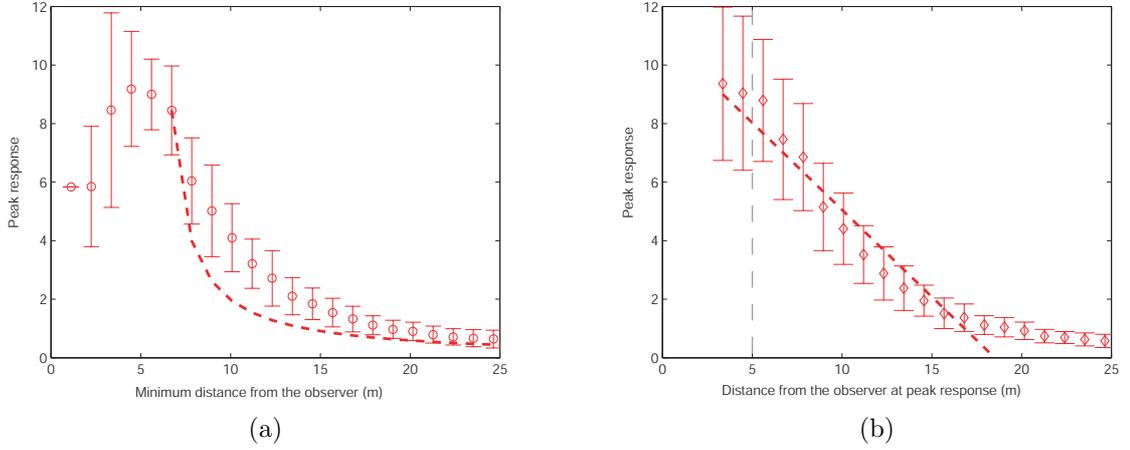


FIGURE 3.9. Mean peak response \bar{R}_{peak} (denoted by circles) and standard deviation σ_R (denoted by error bars) versus the distance from the observer for Rind model with tracking. (a) \bar{R}_{peak} versus the minimum distance from the observer the object ever reached. The dashed line represents a non-linear curve used to fit data as explained in the text. (b) \bar{R}_{peak} versus the distance from the observer at which the peak response was attained. The thick dashed line represents a linear fit as explained in the text. The thin dashed vertical line denotes the distance at which collision is imminent.

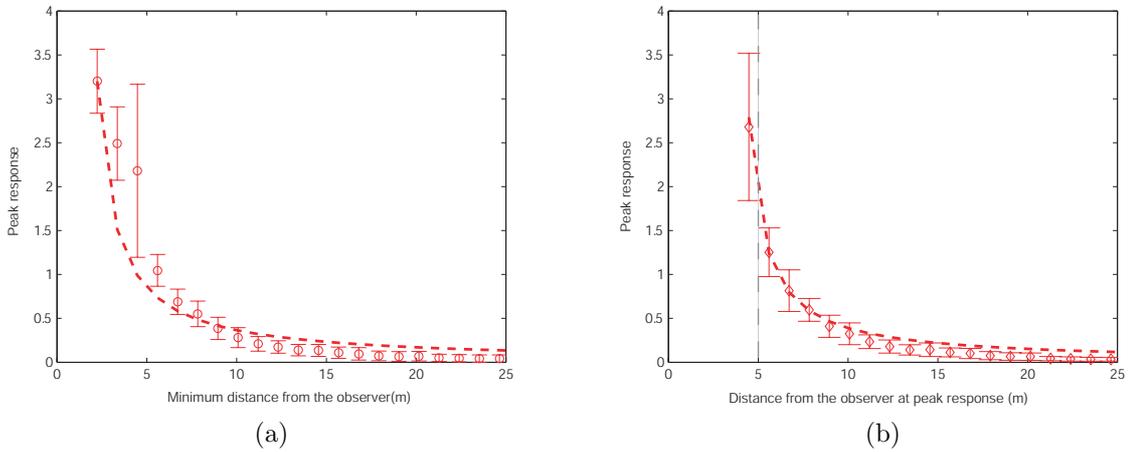


FIGURE 3.10. Mean peak response \bar{R}_{peak} (denoted by circles) and standard deviation σ_R (denoted by error bars) versus the distance from the observer for STI model with tracking. (a) \bar{R}_{peak} versus the minimum distance from the observer the object ever reached. The dashed line represents a non-linear fit to the data. (b) \bar{R}_{peak} versus the distance from the observer at which the peak response was attained. The thick dashed line represents a non-linear fit to the data. The thin dashed vertical line denotes the distance at which collision is imminent. (see text for details)

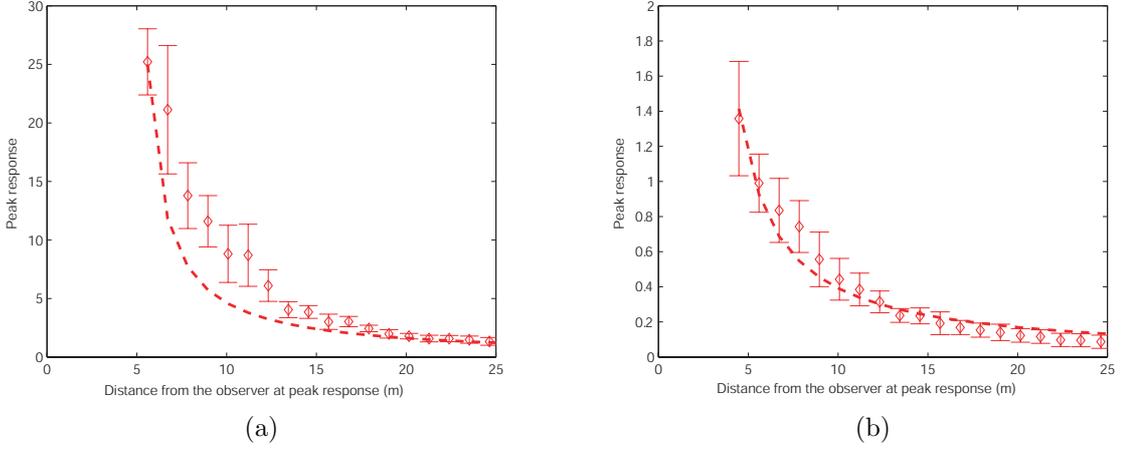


FIGURE 3.11. Mean peak response \bar{R}_{peak} (denoted by circle) and standard deviation σ_R (denoted by error bar) versus the distance from the sensor plane for noisy tracking. The dashed line in both the cases represents respective non-linear fits as described in the text. (a) \bar{R}_{peak} versus the distance from the observer at which the peak response was attained for the Rind model with a noisy tracking algorithm. (b) \bar{R}_{peak} versus the distance from the observer at which the peak response was attained for the STI model with a noisy tracking algorithm.

In the next section, we test collision models for the case of noisy tracking.

3.6.5 Effect of noisy tracking

In all the simulations with tracking discussed above, the tracking algorithm was able to exactly compute the position of the approaching object. This might not necessarily be achievable in a complex real-world scenarios with a non-ideal tracking algorithm. To test whether a less accurate tracking algorithm would be as effective as an exact algorithm, we introduced measurement noise in the centroid computation as detailed in Section 3.5. The results from this set of simulations for both the Rind and the STI model with tracking are shown in Figure 3.11a and 3.11b, respectively. The plots indicate that the imprecise tracking algorithm was still sufficient to make the response of the collision avoidance algorithms behave almost monotonically with the distance from the object. The \bar{R}_{peak} values for the Rind model with tracking in this case almost follow an inverse relation with distance as per Equation 3.18, with fitting parameters $K_1 = 25$ and $d_0 = 4.6$ m. For the STI model with tracking, the \bar{R}_{peak} also follows the same equation with fitting parameters $K_1 = 3$ and $d_0 = 2.36$ m. We note that the STI model with tracking is more affected by the noisy tracking algorithm than the Rind model with tracking. This is expected because the response of the STI model with tracking relies totally on the tracking parameters to cancel the effect of motion parallel to the sensor plane. The Rind model with tracking on the other hand has a local inhibitory network that can suppress weak horizontal motion due to noise in tracking.

3.7 Summary

In this chapter, we developed a mathematical framework to analyze the limitations of the collision avoidance models described in Chapter 2. The major shortcoming of any collision avoidance model is due to an inherent limitation of representing three-dimensional motion in the real world on a two-dimensional sensor array. The analysis presented here shows that if the response of collision

avoidance models can be made dependent on the distance of an obstacle from the observer, this problem may be solved. We have shown how tracking may be used with the Rind and STI models to *indirectly* relate their responses with distance. We have also shown that the STI model with tracking performs much better than the Rind model with tracking for an ideal tracking algorithm. However, the performance of the Rind model with tracking was comparable with the STI model with tracking when a noisy tracking algorithm was used. We have also shown that the improvement in the performance of the models with tracking capability is significant as compared to the models without tracking, even in the case of imprecise tracking.

In the next chapter, we present a camera-based physical system that implements the STI and Rind models with tracking. We also compare the performance of the two algorithms with their non-tracking counterparts.

CHAPTER 4

CAMERA-BASED IMPLEMENTATION OF COLLISION AVOIDANCE
MODELS

In the previous chapter, we described Rind and STI models with tracking capability that may be used to compute a measure of the distance from an obstacle to an observer. However, the Rind and STI models with tracking capability were simulated for very simple scenarios (a white obstacle approaching in a non-textured black background). In this chapter, we test the validity of the same models through an implementation of a camera-based system that utilizes a tracking algorithm to detect collisions with a physical object. The physical implementation of the models is important because we can never fully simulate all aspects of a true collision. The change in the contrast of an approaching object due to its orientation with respect to a light source, or the effect of non-uniform motion of an approaching object, for instance, are difficult to model in an artificial environment. Thus, a system implementation of the algorithms helps us understand the problem better and make necessary changes to improve upon the existing models.

We have also implemented the STI and Rind models without tracking as base algorithms with which we compare our test results. Experimental results for a variety of collision and non-collision scenarios are described for all four algorithms.

4.1 System details

In this section, we describe the physical system that was developed to test the Rind and STI models. All algorithms were tested for collision and non-collision scenarios. The tracking-capable collision detection algorithms were implemented so as to keep the approaching object in the center of the visual field. The setup of the experiment is shown in Figure 4.1. The hardware comprises a web-camera (Labtec webcam 6.0.1) as a visual sensor, a servo motor (Futaba HS700BB) connected at the base of the webcam for tracking, a pulley and DC motor arrangement from which the obstacle is suspended, and an Acer laptop (Microsoft Windows XP OS, 1.66 GHz, dual-core Intel Centrino processor) for processing the algorithms. The collision detection and tracking algorithms were implemented in software using the Visual C++ package (Visual Studio, Microsoft).

The setup shows a DC motor and pulley arrangement that allows for the repeatability of experiments and was used to control the speed at which the object translates in the visual field. This was done by adjusting a variable power supply which controlled the rotational speed of the pulleys. The object was suspended from a string which went around the two pulleys (see Figure 4.1b). The motion of the string added an additional swinging movement (parallel to the camera plane) to the object which was random in nature. This made each run of an experiment slightly different even when the speed of the servo and pulley system were the same. The swinging motion also affected the orientation of the object with respect to the ceiling lights that were used for illumination. This in turn affected the contrast of the object as it moved and added more complexity and randomness to each experimental run.

The collision detection system uses a webcam as a visual sensor. The field of view of the camera was $\pm 23^\circ$ about an axis normal to its sensor plane. The webcam can operate at up to 20 frames per second (fps) while capturing images of size 352×288 . In order to process images at 15 fps, it was necessary to downsample each frame by a factor of two such that the algorithms operated on an image size of 176×144 .

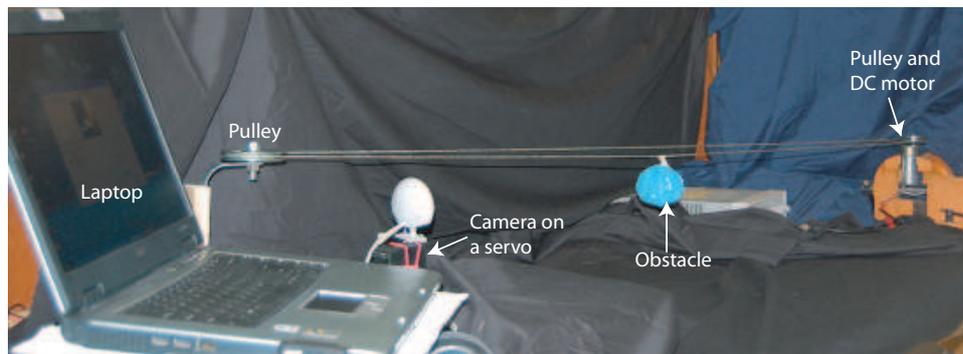
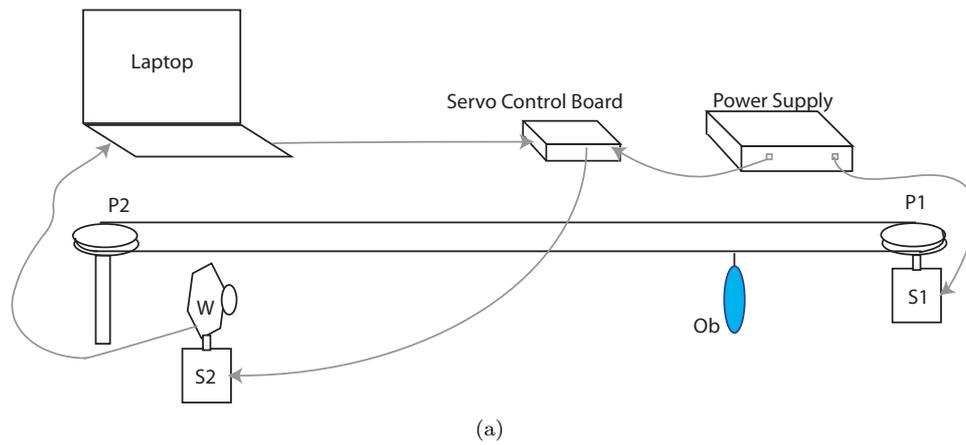


FIGURE 4.1. Experimental setup of the collision detection experiment. (a) Diagram of the setup. Two pulleys (P1 and P2) control the trajectory of the obstacle (Ob), a blue colored disc. Pulley P1 was mounted on a DC motor (S1) and the speed of the obstacle was set by varying the voltage of a power supply. A webcam (W) mounted on a servo motor (S2) sent visual input to a laptop. The laptop runs custom developed software to process the visual information and detect an impending collision. For the algorithms with tracking, the laptop sent control signals to the servo motor S2 via a servo control board. (b) Photograph of the setup.

We implemented a simple tracking algorithm to follow a specifically colored (blue) object by using a centroid detection scheme. A more sophisticated tracking algorithm could be used to tackle more general scenarios (Higgins and Pant, 2004b). The simple algorithm is beneficial in that it does not critically affect the computational speed of the system, and still provides reasonable performance to compare the Rind and STI algorithms with and without tracking capability. The algorithm tracks an object by utilizing a servo motor connected at the webcam's base. This single servo motor allows for only one degree of freedom for the webcam: rotation about its vertical axis. The input frame from the webcam has three color channels: red, green, and blue. We used only the blue color plane to track the object within the frame. The object itself was chosen to be a bright blue colored disc. Since the webcam can only rotate about its vertical axis, the tracking algorithm computed a positional parameter based on the centroid of the blue pixels in the blue channel of every frame. The centroid was computed by using the following formula:

$$x_c = \frac{\sum_i x_i \cdot \sum_j T(I(x_i, y_j))}{\sum_i \sum_j T(I(x_i, y_j))} \quad (4.1)$$

where $I(x_i, y_j)$ is the intensity of the blue pixel at image location (x_i, y_j) , and

$$T(u) = \begin{cases} u & u > 0.95I_{max} \\ 0 & u \leq 0.95I_{max} \end{cases} \quad (4.2)$$

with I_{max} as the maximum intensity in the entire blue channel of the image.

This centroid value was then processed by a PD (proportional-derivative) controller treating $x_e = (x_c - x_m)$ as an error signal, where x_m is the center of the image along the horizontal axis and is considered the origin. The differential error signal \dot{x}_e was generated by taking the difference between the error signal of the current and previous frame. The output signal S_{cnt} was generated by using the following relationship:

$$S_{cnt} = K_P \cdot x_e + K_D \cdot \dot{x}_e \quad (4.3)$$

where K_P and K_D are proportional and derivative gain constants, respectively. The values of K_P and K_D were experimentally determined. A servo control board (USB 16-servo controller, Pololu Corporation) was used to send this signal to the servo motor that controlled the orientation of the webcam. The response of the algorithms with tracking were affected by the frame rate at which the webcam captured images. A frame-rate of 15 fps was not sufficient to show large motions near the camera in a smooth manner. This caused large adjustment in the camera angle to track the object. This jerkiness in the motion of the camera sometimes has a negative effect on the response of the algorithms with tracking.

A snapshot of the graphical user interface (GUI) used to control the experiments is shown in Figure 4.2. The software allows for the response of the algorithms to be recorded either at each frame or as the peak of a running average of the algorithm's response during any experiment. The length of the running average window was set to three. This peak running-average was used to compare the performance of the algorithms.

4.2 Results

In this section, we present results from experiments that demonstrate the effectiveness of tracking in collision avoidance. A comparison is made between the algorithms with and without tracking.

4.2.1 The STI algorithm

The STI algorithm with and without tracking was evaluated under the same test conditions. The object (blue disc) was made to move at a speed of 17 cm/sec. The flat surface of the disc was made

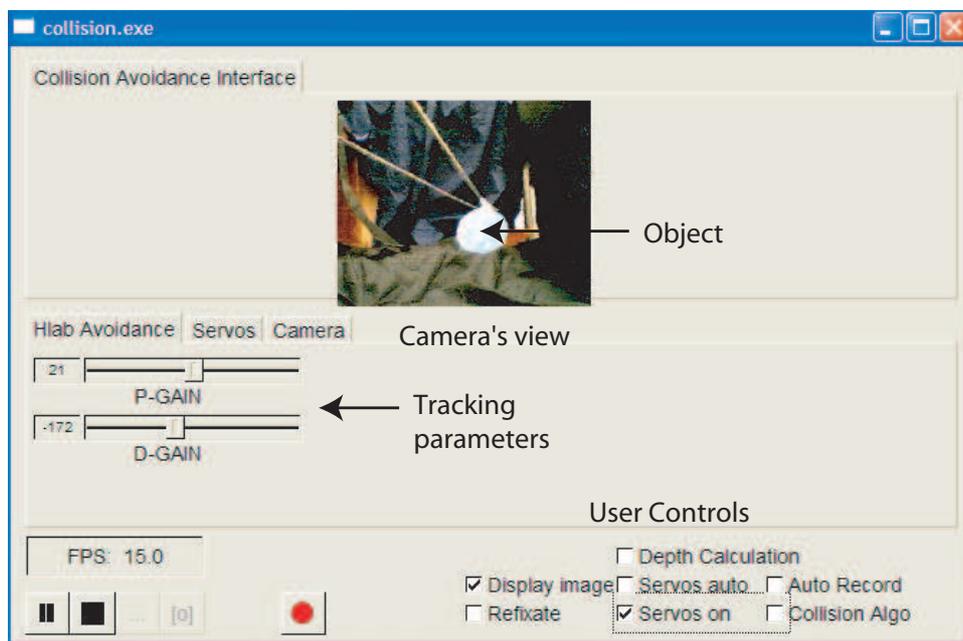


FIGURE 4.2. A snapshot of the GUI for the collision detection algorithms. The webcam's image is displayed in the center and shows the approaching obstacle. The parameters for the tracking algorithm may be adjusted using the sliding bars. The user has control over when to start recording data and which algorithm to select based on a combination of tick-boxes.

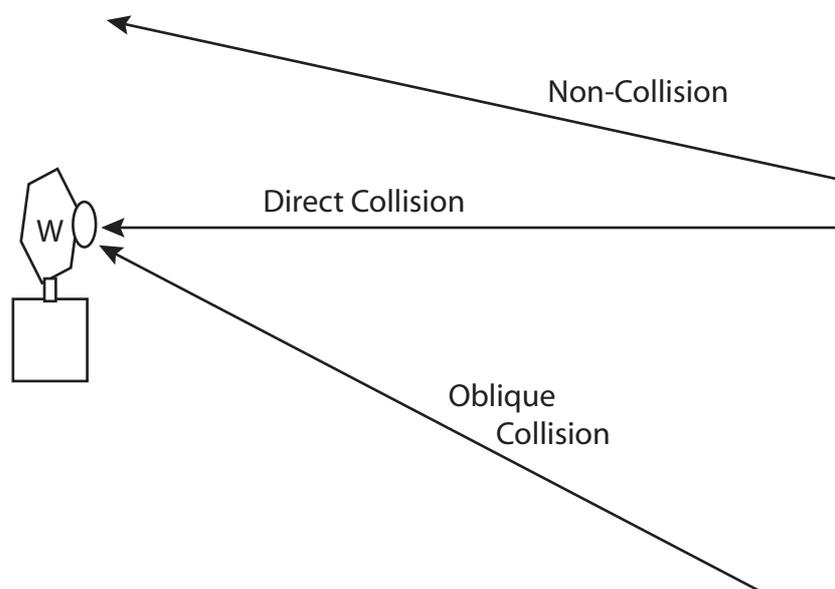


FIGURE 4.3. Different approach trajectories of an object. The lines indicate a direct collision, an oblique collision, and a non-collision approach towards a webcam (W).

to face the webcam at the start of every experiment. However, any small change in the orientation due to the self-motion of the object was not corrected for. We first present response traces for typical direct-collision, oblique-collision, and non-collision cases as shown in Figure 4.3.

Response traces were computed by taking an average of ten direct-collision approach cases. In Figure 4.4a, the thin line shows the average response of the STI algorithm without tracking. The thick line shows the response of the STI algorithm with tracking. As expected, the responses for both with and without tracking cases are almost the same since the object is almost always in the center of the visual field. The slight difference in the two traces originates due to the random swinging of the object as it approaches the webcam. For the algorithm with tracking, the camera moves to compensate this motion and thereby affects the response. The response peaks at roughly 1.2 seconds before the collision at which instant the object occupies the entire visual field of the webcam. The dashed line in the figure shows the distance of the object from the webcam. Note that the traces show the response even after the object goes behind the webcam.

Figure 4.4b shows the average response for the object approaching at a slightly oblique trajectory. Ten experiments were performed for the STI algorithm with and without tracking and the response was averaged. The angle of approach was 10° off the line normal to the camera plane. Some part of the object was always within the visual field of the camera throughout its approach. The thin trace once again represents the response of the STI algorithm without tracking. The thick trace represents the response of the same algorithm with tracking capability. It is clear that even with a slightly oblique approach, the response of the simple STI algorithm is much smaller as compared to a direct collision approach. However, the response trace for the STI algorithm with tracking is similar to the case of direct collision trajectory. This is because the webcam was able to center the approaching object in its visual field for the STI algorithm with tracking, thereby increasing the net expansive motion seen by the camera. The peak in this case is more sustained because of the camera motion which increased the response of the algorithm with tracking, as explained earlier.

We compare the above responses with the average response to a non-collision case. Again, ten

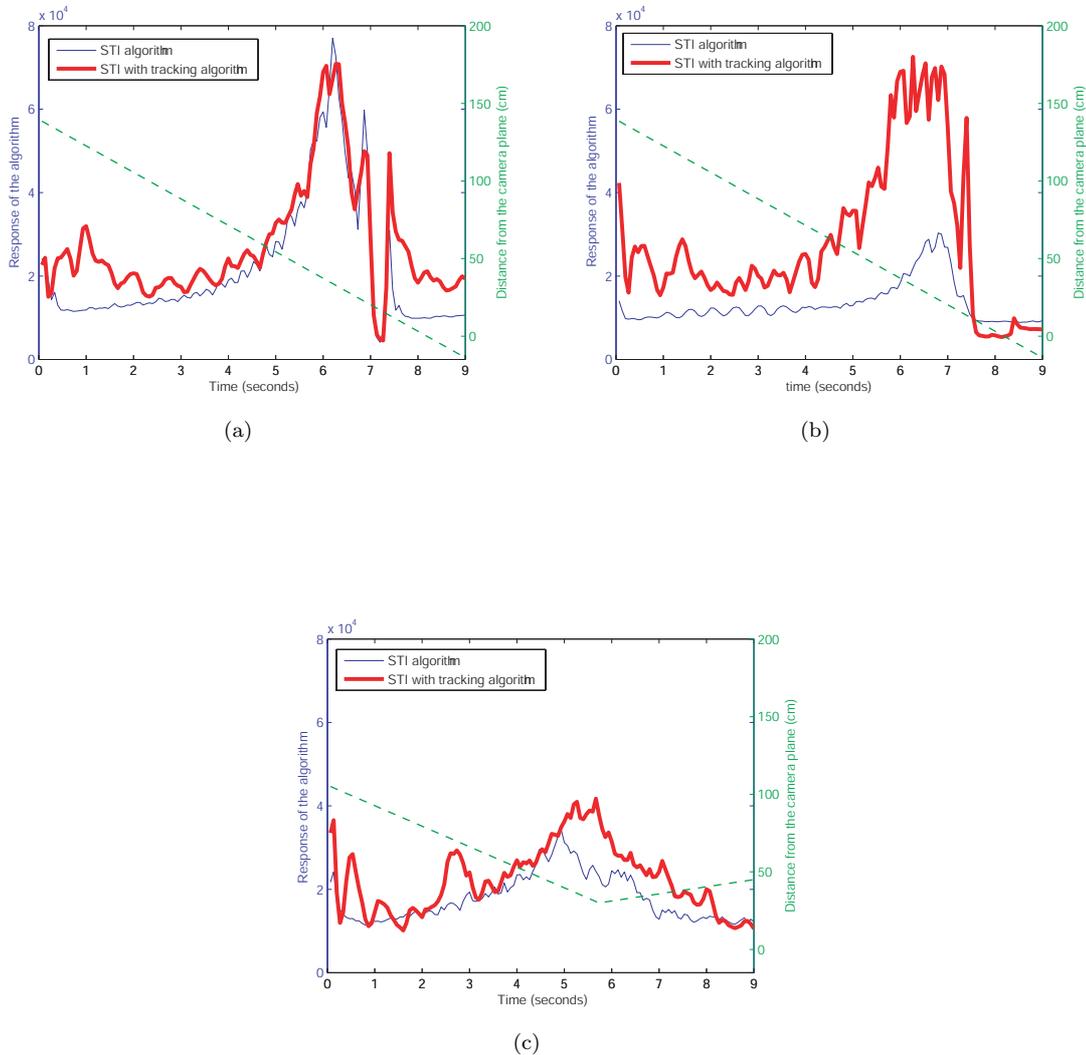


FIGURE 4.4. Response of the STI algorithm with and without tracking averaged over 10 experiments each. The average responses are plotted against time for direct collision, oblique collision, and no collision cases. The thin line represents the frame-by-frame response of the simple STI algorithm. The thick line represents the response of the STI algorithm with tracking. The response magnitude is shown on the left y-axis. The dashed line shows the distance of the obstacle from the camera plane as a function of time. The distance scale is shown on the right y-axis. Negative distances mean that the object is behind the camera. (a) Response for a direct-collision scenario. The response tracks the approach of the obstacle and peaks roughly a 1.2 sec before collision. (b) Response for an oblique approach scenario, with the angle of approach set at 10° . Note that the response of the algorithm with tracking is similar to the direct-collision case, while it is much diminished for the non-tracking STI algorithm. (c) Response for a non-collision scenario. The responses of the algorithm for both with and without tracking cases are significantly smaller than the direct collision cases.

experiments were performed for each case and the response traces were averaged. The trajectory was set such that the object was visible in the view field for virtually the entire length of the experiment as shown in Figure 4.3. It entered the view field from the right side and exited from the left, and was always at a distance of ≥ 30 cm. Figure 4.4c shows the response traces. We find that the response of the STI algorithm with tracking (thick line) is greater in magnitude when compared with the STI algorithm (thin line). However, compared to the responses for direct and oblique collision, these responses were much smaller in magnitude.

In an actual collision avoidance system, the decision to make a course correction has to be made at some point. We have used the peak value from the running average of the response to discriminate collision versus non-collision. The running average was computed for smoothing the raw responses of the algorithm. This peak running-average value may be compared with a threshold to make a decision to turn away from an obstacle when collision is imminent. Here, we compare the STI and STI algorithm with tracking quantitatively by using peak running average as the figure of merit for various collision and non-collision scenarios to answer this question: is it possible to reliably set a threshold to detect the collision cases without raising a false alarm?

Figure 4.5 shows the mean and standard deviation of the peak running-average values recorded for multiple experiments with the STI algorithm with and without tracking. The first pair of bars in the figure show the data for non-collision experiments. Only ten experiments were conducted for each case because the standard deviation of the peak running-average responses was small. The object translated at a speed of 17 cm/sec in a trajectory such that it was always at a distance ≥ 30 cm. The mean response of the STI algorithm with tracking (gray bar) is about twice as large as the response of the non-tracking algorithm (white bar). The response of the algorithm with tracking is larger because of the motion of the camera while tracking.

The second pair of bars show the mean and standard deviation for ten direct-collision experiments each. The responses for both the tracking and without tracking cases are comparable for this set. The mean of the peak running-average values for the direct-collision case is at least three times that for the non-collision case.

The third pair of bars show the mean values of the peak running-average response of the algorithm for an oblique trajectory where the starting position of the object was within $\pm 5^\circ$ of the axis normal to the sensor plane. Each algorithm was tested for 20 experiments because the standard deviation for oblique trajectories was larger than the standard deviation for direct and no collision trajectories. The responses for the algorithm with and without tracking are not too different from each other and are comparable to the mean values of the direct-collision trajectories. In this case, the angle of approach is narrow and during the last phase of approach, the obstacle is almost in the center of the visual field and hence elicits a strong response for both the cases. However, if the angle of approach is slightly increased to be between 8° and 10° off-axis, the mean of the peak running-average values are very different for the tracking and non-tracking cases as seen in the fourth pair of bars in Figure 4.5. Data was collected from 20 experiments for each case. The mean of the peak running-average value for the STI algorithm without tracking reduced by almost half of its value as compared to the case of oblique trajectories within 5° of the normal axis. The mean value of the STI algorithm with tracking, on the other hand, is almost unchanged. This was due to the ability of the algorithm with tracking to keep the object in the center of its visual field, thereby increasing the peak running-average response.

This is also witnessed in the fifth and sixth pair of bars in Figure 4.5. The angle of approach was increased to 15° and 20° , respectively. The mean of the peak running-average value for the STI algorithm without tracking for the latter case is even smaller than the no collision case. This was due to the wide angle of approach of the object at the start of the trajectory. The mean of the peak running-average for the STI algorithm with tracking remains almost unchanged from the direct collision case.

Except for the oblique approach case from wide angles (20°), the mean value of the STI algorithm

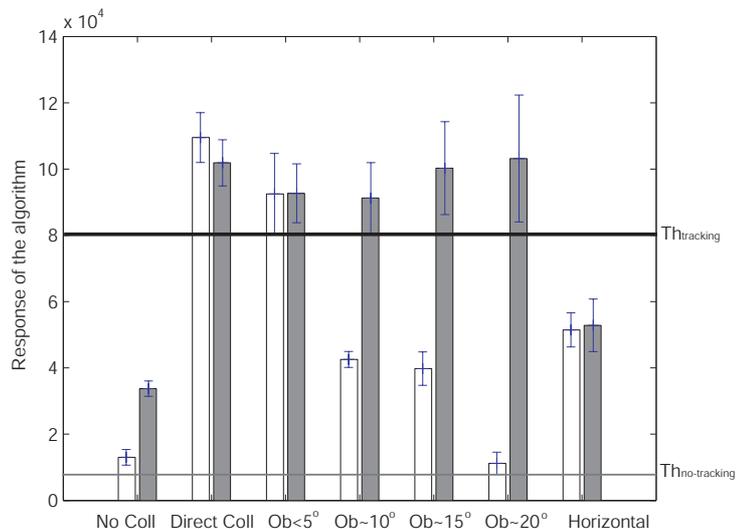


FIGURE 4.5. Experimental data from the STI algorithm with and without tracking for seven scenarios. The white bars show data for the STI algorithm without tracking, while the gray bars show data for the STI algorithm with tracking. Standard deviation is indicated for each experiment. The thick line shows the minimum threshold that needs to be set to detect all collision scenarios for the STI algorithm with tracking, while the thin line shows the minimum threshold for the STI algorithm without tracking. The first pair of bars show the mean response and the standard deviation for non-collision cases. The second pair shows the mean and standard deviation for direct-collision cases. The third pair shows the response for an oblique trajectory where the angle of approach was within $\pm 5^\circ$ of the camera axis. The fourth pair shows the mean response for an oblique approach with the angle of approach between 8° - 10° off the camera axis. The fifth and sixth pair show the responses at 15° and 20° , respectively. The last pair shows the mean and standard deviation of the response for a horizontally moving obstacle.

without tracking is still much larger than its mean for the non-collision cases shown by the first pair of bars in Figure 4.5. This begs a question: is there any other advantage to the STI algorithm with tracking if we can separate most of the collision cases from the non-collision cases by setting a threshold for the STI algorithm without tracking?

To answer this question, we conducted another experiment where the object was made to move horizontally across the visual field at a distance of 20-25 cm from the camera, at a speed of 17 cm/sec. The motion of the object was parallel to the webcam and the object crossed the visual field of the stable camera (in the STI algorithm without tracking) in less than 500 ms. The mean and standard deviation of the peak running-average response for these non-collision experiments is shown by the rightmost pair of bars in Figure 4.5. The responses for the STI algorithm with and without tracking are almost equal in magnitude. The STI algorithm with tracking should have negated the horizontal motion by following the object across the visual field of the webcam. However, due to the fast motion of the object in the visual field, the webcam rotated by large angles. This self-induced motion is responsible for the mean peak response of the STI algorithm with tracking being comparable with its non tracking counterpart. This peak response is about half of the mean peak response for all collision cases shown in the figure. The mean value of the peak running-average response for the STI algorithm, on the other hand, is larger than its mean value for the oblique approach at angles wider than 8° !

This is the defining difference between the algorithm with and without tracking. The STI algorithm without tracking is unable to detect collisions for approach angles larger than 20° and raises a false alarm when an object moves horizontally across the webcam at fast speeds. The thick horizontal line in the figure shows the minimum threshold that is needed to detect all collisions for the STI algorithm with tracking and the thin horizontal line shows the same for the same algorithm without tracking. It is easy to see that the STI algorithm with tracking is able to distinguish all the collision scenarios from the non-collision ones. However, the STI algorithm without tracking cannot do so without raising false alarms. Tracking increases the capability of the STI algorithm to respond reliably to collision scenarios and at the same time, reduce false alarms. We see that even with a simple tracking algorithm the performance of the STI algorithm with tracking improves remarkably and is in agreement with our simulation analysis of Chapter 3.

4.2.2 The Rind algorithm

In this section, we present the results from the Rind algorithm with and without tracking. The experimental setup was the same as that used for the STI algorithm. The speed of the moving object was set at 17 cm/sec. At the beginning of each run, the object was oriented such that its flat surface faced the webcam. The responses were averaged over ten experiments for all the scenarios.

Figure 4.6a shows average response traces for the Rind algorithm with and without tracking in a direct-collision scenario. The response of the simple Rind algorithm (thin line) is very similar to the response of the algorithm with tracking capability (thick line). This is expected since the object is always in the center of the view field of the camera in the case of a direct collision. The main difference between these traces and those in the case of the STI algorithms is that here, the response does not increase until the object is very close to the webcam. This is because the Rind algorithm relies on the edge expansion to overwhelm its lateral inhibitory network. For the speeds that we were able to generate in the experimental setup, this was not true until the object was very near to the webcam. This practical limitation of the Rind algorithm has also been reported by other authors (Cuadri *et al.*, 2005a; Stafford *et al.*, 2007b).

The average response for an object approaching at an oblique trajectory with the starting position at angle of 10° is shown in Figure 4.6b. The object was always in the visual field of the camera throughout its approach. The response of the Rind algorithm (thin trace) is much smaller than that of the Rind algorithm with tracking (thick line). The tracking mechanism is able to keep the approaching object in the center of the visual field. Hence, the response is sufficiently larger with tracking. This again illustrates the importance of tracking in oblique approach scenarios. The response peak for the average is smaller than observed for the direct collision case. This is because the response of the algorithm with tracking peaked at different time instants for different experiments due to the random swaying of the object as it approached. Upon averaging, this made the mean response trace peak at a lower value but with a greater time spread.

Figure 4.6c shows the response curves for a typical non-collision case. The response of the Rind algorithm with tracking (thick line) is comparable with the response of the Rind algorithm without tracking (thin line) and is comfortably smaller than the response in the case of collision cases.

Next, we compare the mean of the peak running-average response for the Rind algorithm with and without tracking for a set of collision and non-collision settings.

The first two pair of bars in Figure 4.7 show the mean and standard deviation of the response of the algorithm with and without tracking for non-collision and direct-collision cases. A total of 40 experiments (10 for each case) were conducted. Each case differed in the starting position of the object (between 140 to 100 cm from the camera) and the instantaneous speed of the object due to its swinging motion, which was random. The mean value of the Rind algorithm without tracking is shown as a white bar, while the response of the Rind algorithm with tracking is shown as a gray bar. As expected, the response of the two algorithms are similar for both the cases. The difference

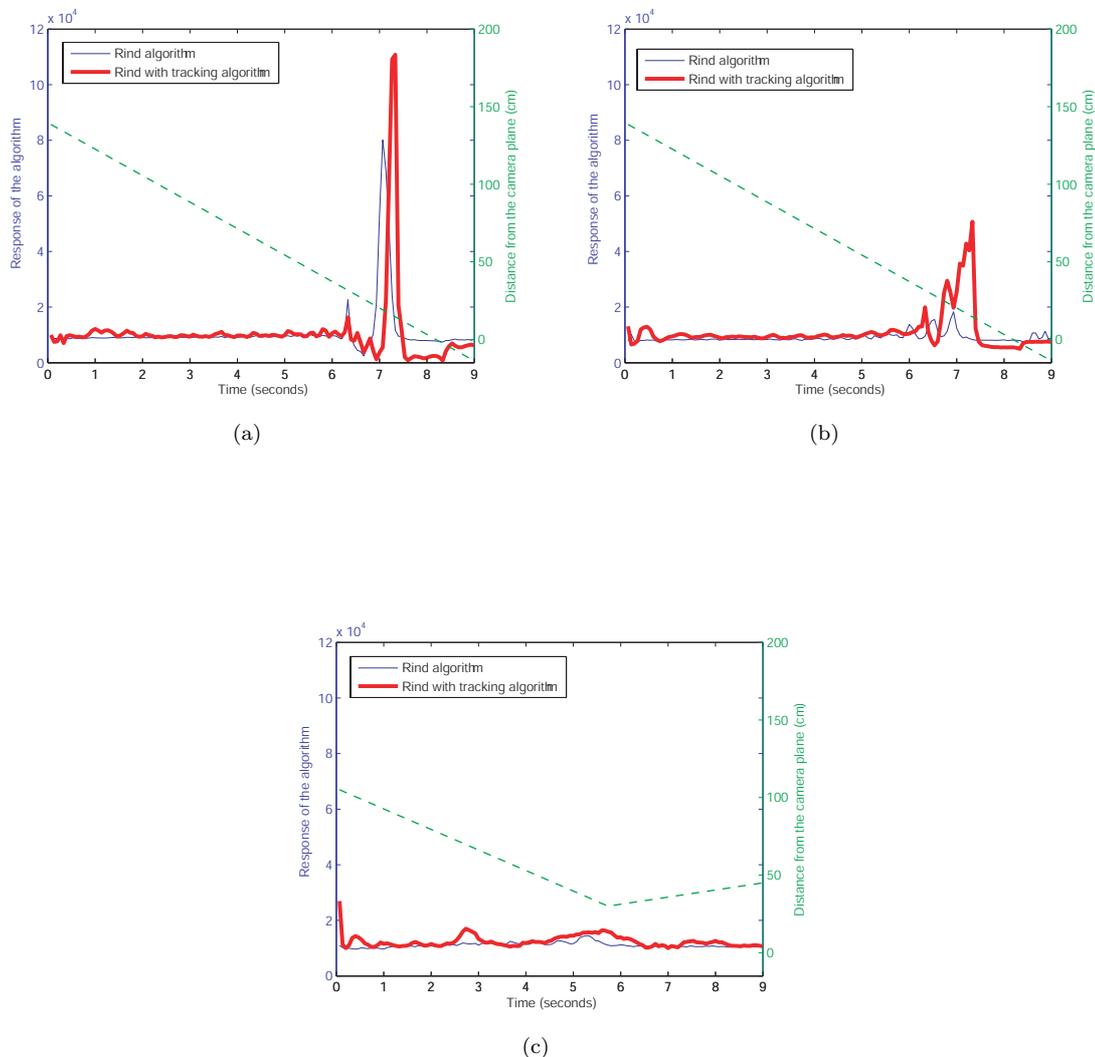


FIGURE 4.6. Response of the Rind algorithm with and without tracking averaged over 10 experiments each. The average responses are plotted against time for direct collision, oblique collision, and no collision cases. The thin line represents the average frame-by-frame response of the Rind algorithm without tracking. The thick line represents the average response of the Rind algorithm with tracking. The response magnitude is shown on the left y-axis. The dashed line shows the distance of the obstacle from the camera plane as a function of time. The distance scale is shown on the right y-axis. Negative distances indicate that the object is behind the webcam. (a) Response for a direct-collision scenario. The response tracks the approach of the obstacle and peaks just before collision. (b) Response for an oblique approach scenario, with the angle of approach set at 10° . Note that the average response of the Rind algorithm with tracking is much larger than the response of the non-tracking algorithm (c) Response for a non-collision scenario. The responses for both the cases are significantly smaller than the direct collision cases.

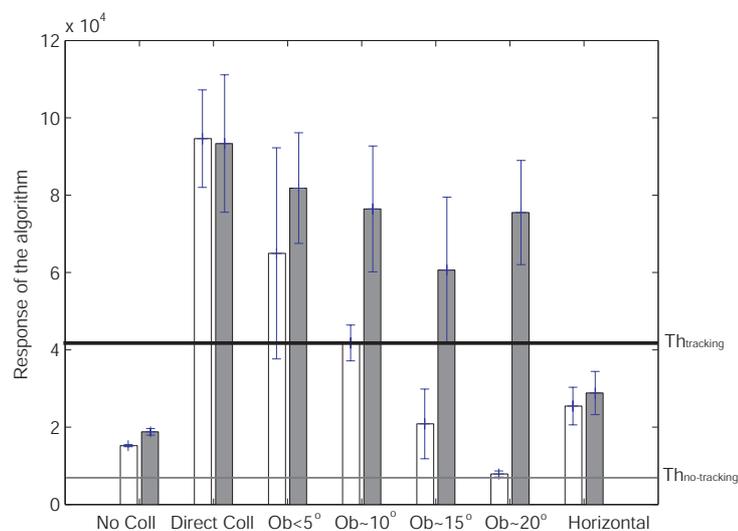


FIGURE 4.7. Experimental data from the Rind algorithm with and without tracking for seven scenarios. The white bars show data for the Rind algorithm without tracking, while the gray bars show data for the Rind algorithm with tracking. Standard deviation is indicated for each experiment. The thick line shows the minimum threshold that needs to be set to detect all collision scenarios for the Rind algorithm with tracking, while the thin line shows the minimum threshold for the Rind algorithm without tracking. The first pair of bars show the mean response and the standard deviation for non-collision cases. The second pair shows the mean and standard deviation for direct-collision cases. The third pair shows the response for an oblique trajectory where the angle of approach was within $\pm 5^\circ$ of the camera axis. The fourth pair shows the mean response for an oblique approach with the angle of approach between 8° - 10° off the camera axis. The fifth and sixth pair show the responses at 15° and 20° , respectively. The last pair shows the mean and standard deviation of the responses for a horizontally moving obstacle.

between the mean value for collision case and the non-collision case is larger than a factor of three for both the algorithms.

The third pair of bars in Figure 4.7 shows the mean value for the oblique collision case when the angle of approach is within 5° of the axis normal to the camera plane. The mean value of the Rind algorithm without tracking is smaller than the direct-collision value by almost 30%. Also, the standard deviation for the Rind algorithm in this case was about two-thirds of its mean value. This is due to the action of the lateral inhibitory network in the Rind model that suppresses slow expansion of the obliquely approaching object. The swinging motion of the object near the camera sometimes caused the peak running-average value to be very high. The response of the Rind algorithm with tracking is larger than the Rind algorithm without tracking by virtue of its keeping the object in the center of the visual field.

The fourth pair of bars in Figure 4.7 shows the mean and standard deviation for an oblique approach at an angle between 8° - 10° . Similar to the STI algorithms, here too, the response of the Rind algorithm without tracking is much weaker than the Rind algorithm with tracking. The response is much smaller here because the Rind algorithm requires sustained increase in activation of its excitatory units to overcome inhibition from its lateral units, which is absent here due to the small increase in the object's size for most of the approach. The response of the Rind algorithm with tracking is not much different from the previous case. The standard deviation for oblique trajectories

was large and depended on the swinging motion of the object.

The fifth and sixth pair of bars show the response of the algorithm for approaches at 15° and 20° angles. The mean of the peak running-average response of the Rind algorithm without tracking decreases progressively as the approach angle increases. The response for approach at 20° angle is smaller than the non-collision case because the object is at the edge of the visual field for virtually the entire trajectory and the increase in the size of the edges is negligible. The mean of the peak running-average response for the Rind algorithm with tracking is not much different for various oblique approaches.

We again compare these oblique-collision responses to that of horizontal motion parallel to the visual field of the camera. The rightmost pair of bars in Figure 4.7 show the response of the two algorithms. The object moved parallel to the image plane at a distance of 20-25 cm from the camera. The mean and standard deviation values for the non-collision horizontal motion are larger than the oblique-collision (angle $\geq 8^\circ$) cases for the Rind algorithm without tracking. The Rind algorithm with tracking has about the same mean value as the Rind algorithm without tracking because of self-induced motion due to large adjustments in the camera angle. However, the mean value is still 50% smaller than in the case of oblique collisions. Thus, a threshold can easily be set for the Rind algorithm with tracking while it is impossible to set a threshold for the simple Rind algorithm without causing false alarms due to horizontal motion parallel to the camera plane. This is shown by the horizontal lines in the figure that represent the minimum threshold that needs to be set to detect all collisions. For the Rind algorithm with tracking, the minimum threshold unambiguously discriminates all collisions from non-collisions (thick line). However, for the Rind algorithm without tracking, distinguishing collisions from non-collisions by setting a simple threshold is impossible.

From our experiments with both the STI and Rind algorithms, we can conclude that tracking clearly improves their collision detection performance. While a threshold is not easy to set for the algorithms without tracking, it is clearly defined in the case of algorithms with tracking, thus increasing the collision detection rate and at the same time, reducing false alarms.

4.3 Summary

The results from various collision and non-collision experiments clearly illustrate that by letting the visual sensor align itself with an approaching object, the effectiveness of the algorithm is dramatically improved. In our study, the performance of the STI algorithm with tracking was better than the Rind algorithm with tracking. This we attribute largely to two factors. Firstly, the Rind algorithm relies on edge detection and in our implementation, the low frame rate of the camera sometimes blurred the motion which made edge detection difficult. Secondly, the lateral inhibitory network in the Rind model *always* tries to suppress the response magnitude of the algorithm. In order to get a strong response from this algorithm, the activation of its excitatory units must be large enough to swamp the effect of inhibitory units. This goes back to the problem of edge detection. Since the frame-rate of our input was not very high, the edge response was weak and so was the response of the excitatory units. The STI algorithm with tracking, on the other hand, does better on both accounts, and in general is an elegant algorithm for collision avoidance.

The primary observation from the experimental results and analysis presented up to this point in the dissertation is that if the relative distance from objects to an observer in a scene are known, it greatly facilitates collision avoidance. We have shown how tracking may be used with the Rind and STI models to *indirectly* compute a measure of distance. This distance may also be computed *directly* from motion parallax by utilizing the speed of objects in a scene. It has been suggested that honeybees use a similar mechanism based on non-directional image speed for obstacle avoidance while navigating through a tunnel (Srinivasan *et al.*, 1993). The image speed is also utilized by honeybees to reduce their speed while landing, and relay information about distances of food sources from hives (Srinivasan *et al.*, 1996, 2000). In the next chapter, we describe two neuronally based non-

directional models that have speed dependent outputs and may be used by artificial systems for obstacle avoidance.

CHAPTER 5

SPEED ESTIMATION BY NON-DIRECTIONAL MOTION MODELS

In the previous chapter, tracking was added to collision avoidance models based on the escape response to detect collisions by indirectly computing object distances. However, escape response based algorithms only provide a reflexive reaction to an imminent collision and therefore, their use may require a sudden change in the orientation and speed of the observer. An autonomously navigating robot is more likely to encounter a scenario where it has to navigate around obstacles that may not be on an imminent collision course. In such cases, it is better to smoothly change the trajectory of the robot. This may be performed by utilizing a direct computation of the relative depth of objects in a scene based on their speed. The speed of an object on the sensor plane may be computed by either solving an ill-posed problem of optical flow or by using a qualitative method such as a spatio-temporal frequency based correlation model. Due to the qualitative equivalence of the two schemes, shown by Verri and Poggio (1989), correlation-based models are more attractive. There is a large body of evidence which indicates that insects also use spatio-temporal frequency based correlation models (Reichardt, 1961; Adelson and Bergen, 1983; Borst and Bahde, 1986; Reichardt *et al.*, 1989). However, these correlation-based models have dissimilar responses to the *same* speed at different spatial frequencies (Srinivasan *et al.*, 1993). Despite this, it has been reported that bees utilize the apparent angular image speed on their compound eyes to perform complicated navigational maneuvers including collision avoidance (Srinivasan *et al.*, 1991, 1997, 2000). Bees match image speed on their left and right eyes to center their flight path when flying between two walls. They hold the image speed on their eyes constant when in flight, and reduce their speed when flying through a tapering tunnel. They use a similar mechanism for smooth landing (Srinivasan *et al.*, 2000). Insect scientists have long speculated that insects may compute the speed from a visual image by either directional motion or non-directional motion computation (Srinivasan *et al.*, 1993; Single and Borst, 1998). Recent experiments have suggested that a non-directional mechanism may be behind this speed estimation (Higgins, 2004; Dacke and Srinivasan, 2007). In this chapter, we present two non-directional speed estimation models derived from the neuronally based EMD model described in Chapter 1. The non-directional models presented here are simplified versions of the earlier models reported by Higgins (2004) and Rivera-Alvidrez (2005). The simplifications that we present here make these non-directional models respond to a broader range of speed and spatial frequencies. This speed sensitive output may then be utilized to compute relative distance of objects in a scene through motion parallax.

5.1 The NDM model

A non-directional multiplication (NDM) based computational model was proposed by Higgins (2004) to explain how insects may compute image speed (see Figure 5.1a). For most correlation-based models, the response is not speed dependent and linear speed response is very difficult to compute. The mean response of the NDM model, on the other hand, has an almost linear relation with speed for a range of spatial frequencies. The approximate linearity with speed may be used to compute the relative depths of objects in a scene, and thereby avoid collisions. However, the basis of the speed dependence of this model was not clear.

In this chapter we explore the mathematical basis of this speed dependence, and consider a simplified NDM model that also responds linearly with speed for a range of spatial frequencies. The simplification to the model comes in the form of removal of the low-pass filter (LPF) units

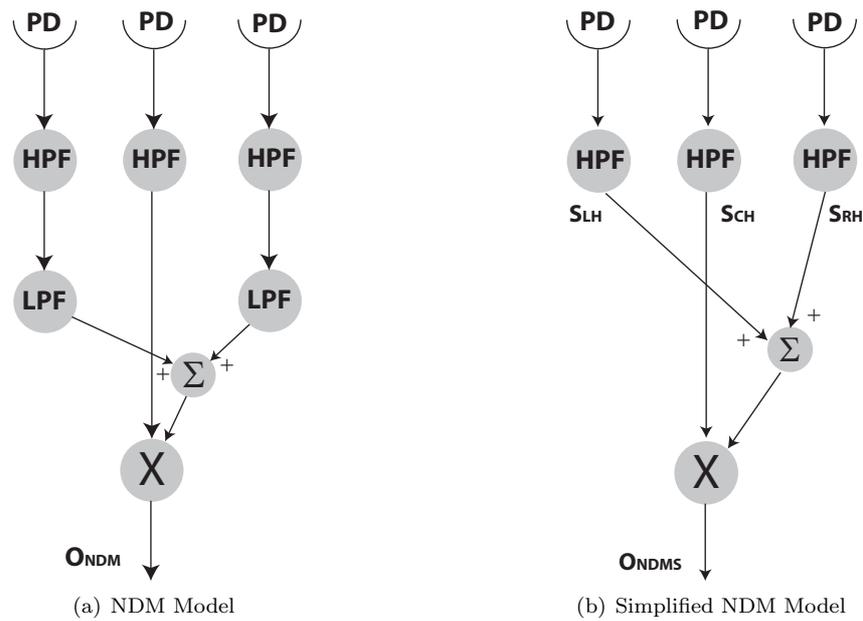


FIGURE 5.1. Non-directional multiplication (NDM) based computational models. (a) Original NDM model. Light intensity is measured by photodetectors (PD) and is then processed by a high-pass filter (HPF) stage. Next, the response of two outer PD's are low-pass filtered (LPF) and added (Σ). The summed output is then multiplied (\times) with the central PD's response. This results in a speed dependent output (O_{NDM}). (b) Simplified non-directional multiplication (Simplified NDM) model. As in the case of the original NDM model, light intensity is measured by photodetectors (PD) and is then processed by a high-pass filter (HPF) stage. In this case, no LPF units are present and the filtered response of two outer PD's is directly added (Σ). It is then multiplied (\times) with the central PD's filtered response. This results in a speed dependent output (O_{NDMS}).

in the original model, as shown in Figure 5.1b. A single simplified NDM unit comprises three photodetectors with high-pass filter (HPF) units. The response from the two lateral HPF units is added and then multiplied with the central HPF unit's response. The spatial summation of the photodetector responses acts like a spatial low-pass filter and together with the temporal high-pass filter, the output is speed dependent.

The response characteristics of a simplified NDM unit for a sinusoidal grating input are described below.

5.1.1 Response characteristics

We studied the response of a simplified NDM unit to a sinusoidal grating

$$I(x, t) = \frac{1}{2} (1 + C \sin(\omega_x \cdot x + \omega_t \cdot t)) \quad (5.1)$$

where C is the contrast, ω_x is the spatial frequency in cycles/space-unit, and ω_t is the temporal frequency in cycles/sec. This grating is offset such that it is strictly positive. The spacing between adjacent photodetectors was set to be Δ . Thus the spatial phase difference between two adjacent photodetectors was $\omega_x \cdot \Delta$. For simplicity of derivation and without loss of generality, we chose $\Delta = 1$. A first order temporal filter was employed as the HPF unit with a frequency-domain phase response $\phi_1(\omega_t)$, and magnitude response

$$h_1 = \frac{\omega_t \cdot \tau}{\sqrt{1 + (\omega_t \cdot \tau)^2}} \quad (5.2)$$

where τ is the time constant of the HPF unit. The three adjacent HPF units shown in Figure 5.1b are separated in space, and therefore their responses have both a spatial and a temporal phase term as shown below:

$$S_{LH} = \frac{C}{2} \cdot h_1 \cdot \sin(\omega_t \cdot t + \phi_1) \quad (5.3)$$

$$S_{CH} = \frac{C}{2} \cdot h_1 \cdot \sin(\omega_t \cdot t + \omega_x + \phi_1) \quad (5.4)$$

$$S_{RH} = \frac{C}{2} \cdot h_1 \cdot \sin(\omega_t \cdot t + 2\omega_x + \phi_1) \quad (5.5)$$

Adding the response from the two lateral units (S_{LH} and S_{RH}) and multiplying it with the central unit S_{CH} gives

$$O_{NDMS} = \frac{C^2}{4} \cdot h_1^2 \cdot \sin(\omega_t \cdot t + \omega_x + \phi_1) \cdot (\sin(\omega_t \cdot t + 2\omega_x + \phi_1) + \sin(\omega_t \cdot t + \phi_1)) \quad (5.6)$$

Using common trigonometric identities and inserting the value of h_1 from Equation 5.2, the temporal mean response of the model can be shown to be

$$\bar{O}_{NDMS} = \frac{C^2}{4} \cdot \frac{(\omega_t \cdot \tau)^2}{1 + (\omega_t \cdot \tau)^2} \cdot \cos(\omega_x) \quad (5.7)$$

This may be compared to the temporal mean response of the original NDM model to the same stimulus as derived by Higgins (2004):

$$\bar{O}_{NDM} = \frac{C^2}{4} \cdot \frac{(\omega_t \cdot \tau)^2}{(1 + (\omega_t \cdot \tau)^2)^2} \cdot \cos(\omega_x) \quad (5.8)$$

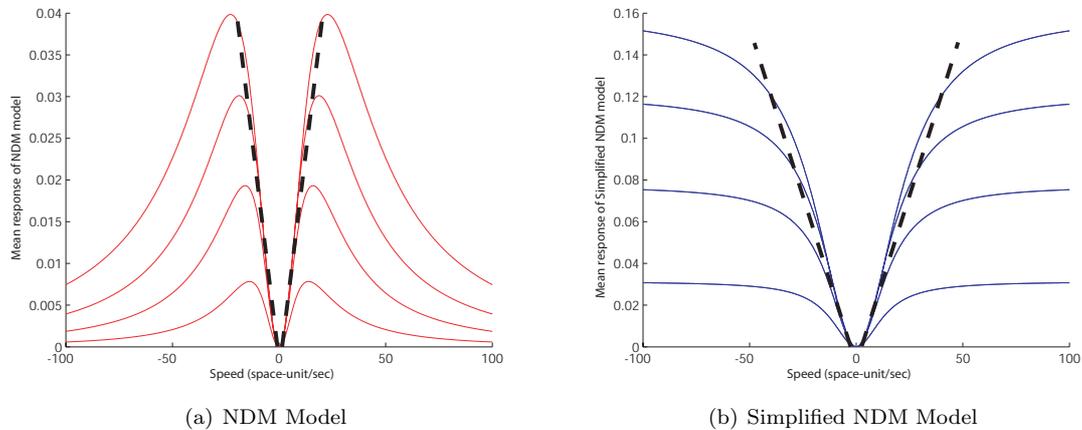


FIGURE 5.2. Response of the NDM models against stimulus speed. (a) Mean response of the original NDM model as the speed of the sinusoidal grating varies from -100 to $+100$ space-unit/sec. Multiple traces show the response at spatial frequencies of 0.08 (maximum peak response), 0.14 , 0.17 , 0.2 , and 0.23 (minimum peak response) cycles per space-unit. (b) Mean response of the simplified NDM model against the speed of a sinusoidal grating. The traces represent the response of the model for the same set of spatial frequencies as in (a). The response is *nearly* linear with speed up to its peak response at which it saturates.

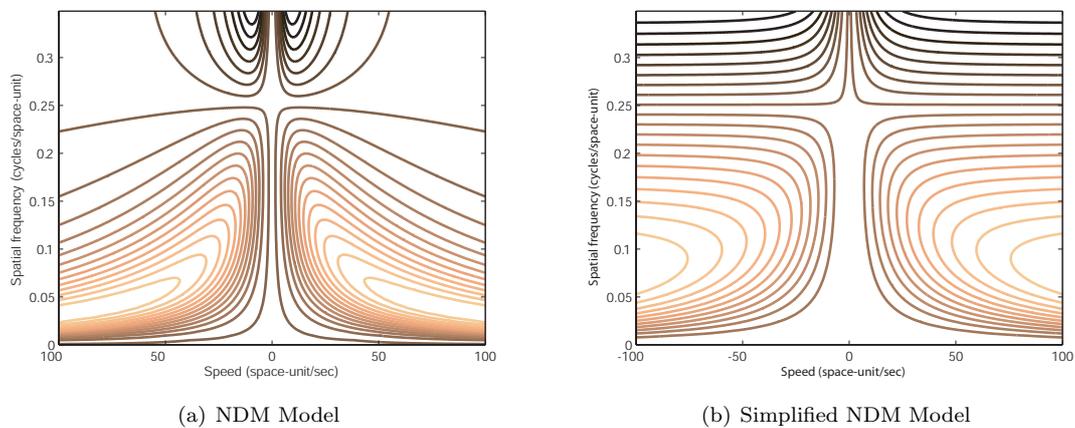


FIGURE 5.3. Contour plots of the mean response of NDM models. (a) Contour representation of the mean response of the original NDM model with varying speed (horizontal axis) and spatial frequency (vertical axis). The curves parallel to the vertical axis show that the response is independent of the spatial frequency in that range. The contour values for spatial frequencies higher than 0.25 cycles/space-unit are negative. (b) Contour plots of the mean response of the simplified NDM model. Notice that the traces become parallel to the horizontal axis in this case as the response saturates at its peak value. The contour values for spatial frequencies higher than 0.25 cycles/space-unit are negative.

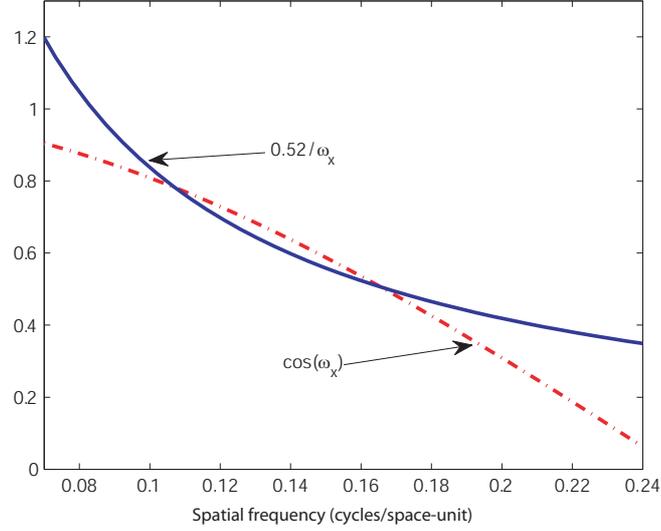


FIGURE 5.4. Comparison between $0.52/\omega_x$ and $\cos(\omega_x)$ for a range of spatial frequencies ω_x .

Figure 5.2 shows the mean response of the original and the simplified NDM models plotted against the speed of a sinusoidal grating stimulus, which is a ratio of the temporal frequency and spatial frequency ($v=\omega_t/\omega_x$). As compared to that of the original model, the mean response of the simplified model saturates at a higher speed as seen in the figure.

Figure 5.3 shows the mean response of the models plotted as contour lines when the spatial frequency and the speed of the stimulus were varied. The contours in the figure that are parallel to the vertical axis represent a spatial frequency independent response. Again, the results show that in comparison to the original model, the simplified model saturates at higher speeds against spatial frequencies as evidenced by the vertical lines in the plot. Note that in the case of the simplified NDM model, the lines become parallel to the horizontal axis. Thus, unlike the original NDM model response which peaks and then falls, the simplified model's response peaks and saturates. This results in an unambiguous representation of speed up to the maximum response of the model. This is a distinct advantage over the original NDM model whose response level may be the same at two speeds owing to the peaking and falling of its mean response.

We have investigated the mathematical basis of this speed dependence which was not clear in the original model. The expression for the mean response of the simplified NDM model (Equation 5.7) has ω_t and $\cos(\omega_x)$ terms. We have plotted $1/\omega_x$ and $\cos(\omega_x)$ against the spatial frequency ω_x in Figure 5.4. We find that the slopes of these two curves for a range of spatial frequencies are roughly the same. Thus, for this range we can replace $\cos(\omega_x)$ with $1/\omega_x$. We can then isolate an ω_t/ω_x term (the stimulus speed) from the expression, such that it becomes

$$\bar{O}_{NDMS} \approx \frac{C^2}{4} \cdot \left(\frac{\omega_t}{\omega_x} \right) \cdot \frac{\omega_t \cdot \tau^2}{1 + (\omega_t \cdot \tau)^2} \quad (5.9)$$

At low temporal frequencies, $(\omega_t \cdot \tau)^2 \ll 1$ and therefore the output varies as a function of (ω_t^2/ω_x) which increases superlinearly with speed (ω_t/ω_x). As the temporal frequency increases towards $\omega_t \cdot \tau \approx 1$, the output becomes proportional to $1/\omega_x$ and saturates at different levels based on the spatial frequency (see Figure 5.2b). Thus, we see that for a range of spatial and temporal

frequencies, the mean response of the simplified NDM model varies monotonically with speed.

5.1.2 Flicker response

For reliable collision avoidance, we would prefer that the model responds to actual stimulus motion and not simply to a sudden change in the light intensity. Ideally, the response of a motion model to flicker should be much weaker than its response to a regularly moving pattern. We investigated this by considering a counterphase sinusoidal flicker stimulus

$$I_f(x, t) = \frac{1}{2} \cdot (1 + C \cdot \sin(\omega_f \cdot t) \sin(\omega_x \cdot x)) \quad (5.10)$$

where ω_f is the flicker frequency in cycles/sec and ω_x is the spatial frequency in cycles/space-unit. Following the derivation of Higgins (2004), the mean output of the model for this stimulus can be shown to be

$$\bar{O}_{NDMS_f} = \frac{C^2}{4} \cdot \frac{(\omega_f \cdot \tau)^2}{1 + (\omega_f \cdot \tau)^2} \cdot \cos(\omega_x) \cdot \sin^2((p_0 + 1)\omega_x) \quad (5.11)$$

where p_0 is the position of the leftmost photodetector unit. Since $\sin^2((p_0 + 1)\omega_x) \leq 1$, therefore,

$$\bar{O}_{NDMS_f} \leq \bar{O}_{NDMS} \quad (5.12)$$

For the original NDM model, the response to the same flicker stimulus may be shown to be (Higgins, 2004)

$$\bar{O}_{NDM_f} = \frac{C^2}{4} \cdot \frac{(\omega_f \cdot \tau)^2}{(1 + (\omega_f \cdot \tau)^2)^2} \cdot \cos(\omega_x) \cdot \sin^2((p_0 + 1)\omega_x) \quad (5.13)$$

The ratio of the response of the model between stimulus motion and flicker is the same as that for the simplified NDM model. The response of the original model is no stronger than the motion response and if we average the response over space, the flicker response is always weaker than the motion response. Much like the original NDM model, if we average the response of several simplified NDM units over space, their mean response will also be stronger to motion than flicker.

5.2 The NDS model

Rivera-Alvidrez (2005) proposed a non-directional summation (NDS) model that was derived from the neuronally based EMD model (Higgins *et al.*, 2004) described in Chapter 1. Instead of a multiplication unit, this model has a summation unit (see Figure 5.5a) that is comparable with the transmedullary neuron Tm1 in the medulla of the fly brain. We have simplified this model by removing the LPF unit from the lateral photodetector paths: the same simplification as in the NDM model. As shown in Figure 5.5b, the photodetector signals are processed through high-pass filter units and then added together. The amplitude response of this unit is speed dependent for a range of spatial frequencies.

Next, we describe the response characteristics of the simplified NDS model for a sinusoidal grating input.

5.2.1 Response characteristics

We study the response characteristics of a simplified NDS unit by deriving its response to a sinusoidal grating $I(x, t)$ given by Equation 5.1. The HPF unit is implemented as a first order high-pass filter and its frequency-domain magnitude response is the same as given by Equation 5.2. The output responses of the HPF units are the same as in Equations 5.3, 5.4, and 5.5. Summing the responses from the three HPF units, we get

$$O_{NDSS} = \frac{C}{2} \cdot h_1 \cdot \sin(\omega_t t + \omega_x + \phi_1) \cdot (\sin(\omega_t t + \phi_1) + \sin(\omega_t t + 2\omega_x + \phi_1)) \quad (5.14)$$

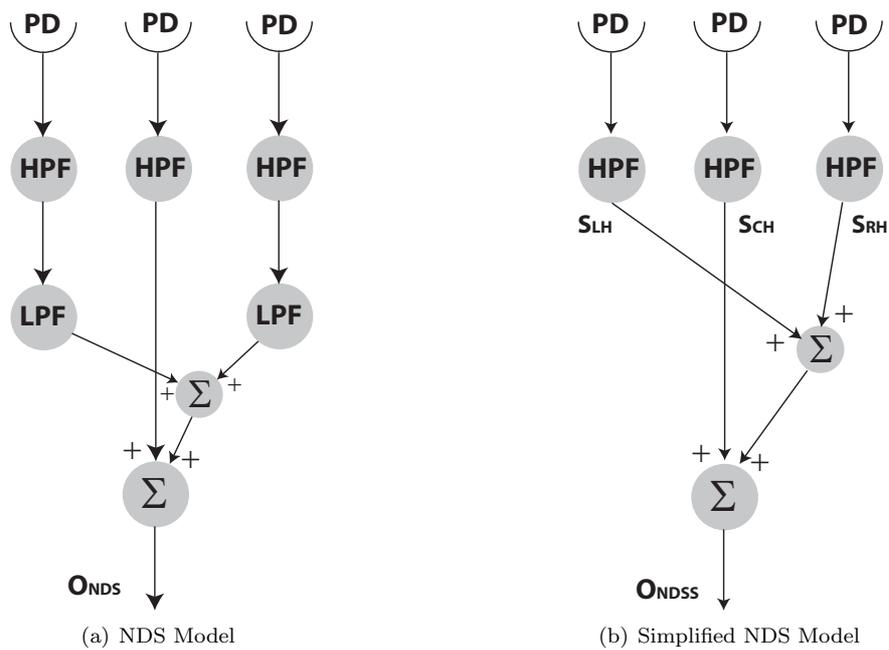


FIGURE 5.5. Non-directional summation (NDS) based computational models. (a) Original NDS model. Light intensity is measured by photodetectors (PD) and is then processed by a high-pass filter (HPF) stage. Next, the response of two outer PD's are low-pass filtered (LPF) and added (Σ). The summed output is then added (Σ) with the central PD's response. (b) Simplified Non-directional summation (NDS) model. As in the case of the original model, light intensity is measured by photodetectors (PD) and is then processed by a high-pass filter (HPF) stage. In this case, no LPF units are present and the filtered response of two outer PD's are directly added (Σ). The summed output is then added (Σ) with the central PD's filtered response.

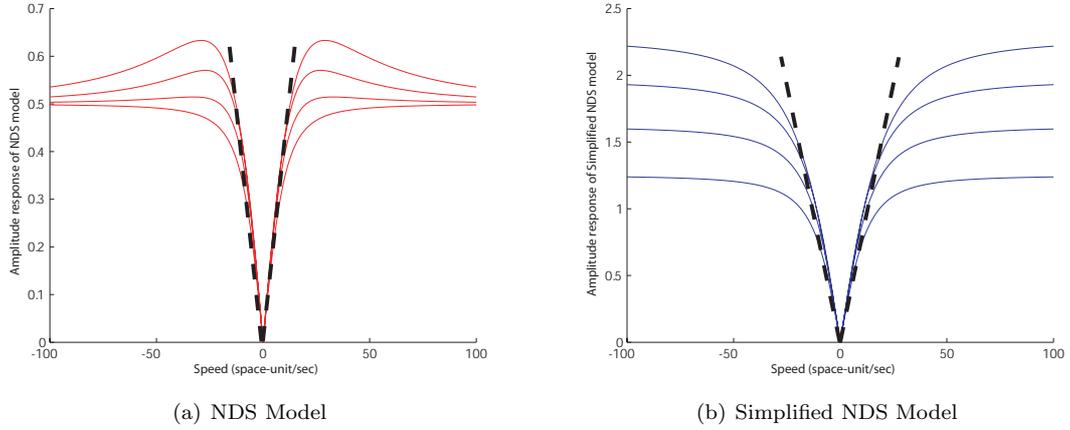


FIGURE 5.6. Amplitude response of the NDS models against stimulus speed. (a) The original NDS model. Multiple traces represent the response of the model at spatial frequencies of 0.08 (maximum peak response), 0.14, 0.17, 0.2, and 0.23 (minimum peak response) cycles per space-unit. (b) The simplified NDS model. The curves represent the response of the model at multiple spatial frequencies same as those in (a). The response of the model is *nearly* linear with speed up to its maximum response at which it saturates.

Simplifying and taking the amplitude of the above expression, we get

$$\hat{O}_{NDSS} = \frac{C}{2} \cdot \frac{\omega_t \cdot \tau}{\sqrt{1 + (\omega_t \cdot \tau)^2}} \cdot (1 + 2 \cos(\omega_x)) \quad (5.15)$$

This may be compared to the amplitude response of the original NDS model to the same stimulus as derived by Rivera-Alvidrez (2005):

$$\hat{O}_{NDS} = \frac{C}{2} \cdot \frac{\omega_t \cdot \tau}{1 + (\omega_t \cdot \tau)^2} \cdot \sqrt{4[\cos^2(\omega_x) + \cos(\omega_x)] + (\omega_t \cdot \tau)^2 + 1} \quad (5.16)$$

Figure 5.6 shows the amplitude response curves at different spatial frequencies against the speed of the stimulus for both the original and the simplified NDS models. It is easy to see that the response of the simplified model (Figure 5.6b) saturates at higher speeds as compared to the response of the original NDS model. Figure 5.7 shows contour plots of the amplitude against the spatial frequency and the speed of the sinusoidal grating. The amplitude response of the original NDS model at high spatial frequencies is highly nonlinear and non-monotonic. This is because the square-root term in Equation 5.16 depends nonlinearly on the cosine of the spatial frequency. The response of the simplified unit, on the other hand, increases linearly with speed (contour lines parallel to the vertical axis in Figure 5.7b) and saturates at its peak value as seen by the contour lines parallel to the horizontal axis.

As in the case of the simplified NDM model, here too the response is nearly linear with speed. This is due to the slope of $1/\omega_x$ and $(1 + 2 \cos(\omega_x))$ being quantitatively close to each other as shown in Figure 5.8. Therefore, for a certain range of spatial frequency we can replace $(1 + 2 \cos(\omega_x))$ by $1/\omega_x$ and extract the speed term from Equation 5.15 to get

$$\hat{O}_{NDSS} \approx \frac{C}{2} \cdot \left(\frac{\omega_t}{\omega_x} \right) \cdot \frac{\tau}{\sqrt{1 + (\omega_t \cdot \tau)^2}} \quad (5.17)$$

With this approximation, for low temporal frequencies we have $(\omega_t \cdot \tau)^2 \ll 1$, and therefore the output is directly proportional to speed (ω_t/ω_x) . As the temporal frequency increases so that

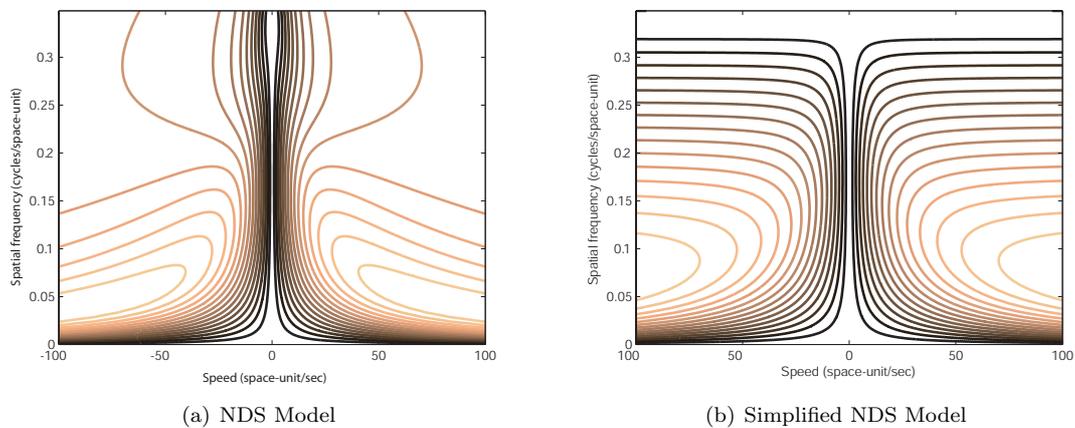


FIGURE 5.7. Contour plots of the amplitude response of NDS models with varying speed (horizontal axis) and spatial frequency (vertical axis). (a) The original NDS model. The contours parallel to the vertical axis show that the response is independent of the spatial frequency in that range. (b) The simplified NDS model. Notice that the traces become parallel to the horizontal axis in this case as the response saturates at its peak value.

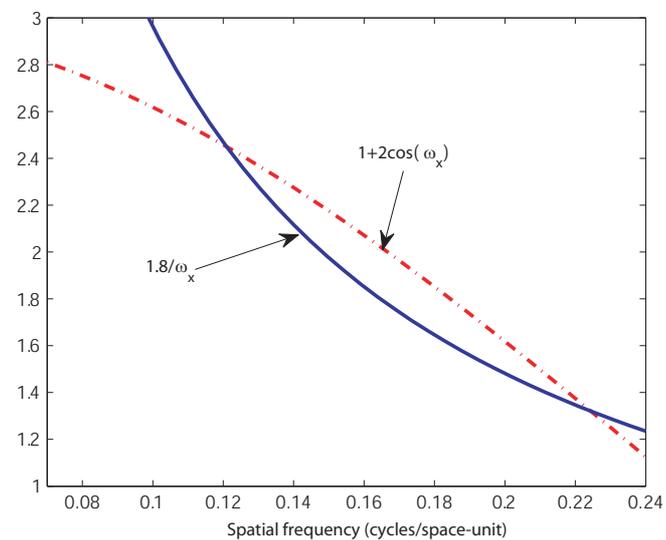


FIGURE 5.8. Comparison between $1.8/\omega_x$ and $(1 + 2 \cos(\omega_x))$ for a range of spatial frequencies (ω_x).

$\omega_t \cdot \tau \gg 1$, the output of the simplified NDS model depends only on the spatial frequency term $1/\omega_x$, and therefore saturates at a different level based on the spatial frequency of the stimulus. Thus the simplified NDS model has a more linear response to speed than the simplified NDM model.

5.2.2 Flicker response

The amplitude response of the simplified NDS model to a counterphase flicker stimulus as shown in Equation 5.10, with the leftmost photodetector positioned at p_0 , is

$$\hat{O}_{NDSS_f} = C \cdot \frac{(\omega_f \cdot \tau)}{\sqrt{1 + (\omega_f \cdot \tau)^2}} \cdot (1 + 2 \cos((p_0 + 1)\omega_x)) \cdot \sin((p_0 + 1)\omega_x) \quad (5.18)$$

Comparing this response to the motion response shown in Equation 5.15, we have

$$\hat{O}_{NDSS} \leq \hat{O}_{NDSS_f} \quad (5.19)$$

The response of the original NDS model to the same flicker stimulus may be shown as (Rivera-Alvidrez, 2005)

$$\hat{O}_{NDS_f} = \frac{C}{2} \cdot \frac{(\omega_f \cdot \tau)}{1 + (\omega_f \cdot \tau)^2} \cdot \sqrt{4[\cos^2(\omega_x) + \cos(\omega_x)] + (\omega_f \cdot \tau)^2 + 1} \cdot \sin((p_0 + 1)\omega_x) \quad (5.20)$$

This response is weaker than the motion response shown in Equation 5.16 by a factor of $\sin((p_0 + 1)\omega_x)$, the same ratio as in the case of the simplified model.

A comparison between the counterphase flicker and sinusoidal grating responses for both the simplified NDS and NDM models is shown in Figure 5.9. The flicker responses in both the simplified NDM and NDS models are considerably weaker than the motion responses. As compared to the original models, the response of both the simplified NDM and NDS models have the same ratio between motion and flicker.

5.3 Summary

The simplifications we have made to the original NDM and NDS models make their responses speed-sensitive up to a higher speed without sacrificing their ability to discriminate motion from flicker. The response also saturates at its peak value, and thus provides an unambiguous representation of speed. The speed-dependent output of the simplified non-directional models may be used in artificial systems to estimate relative depth based on motion parallax. This relative depth information may be used by a collision avoidance model to detect collisions more reliably.

For a real-time application, a software based implementation of these models may not be efficient. It may be more useful to design a chip that computes the image speed in real-time. In the next chapter, we present an analog VLSI implementation of these models and show their characterization results.

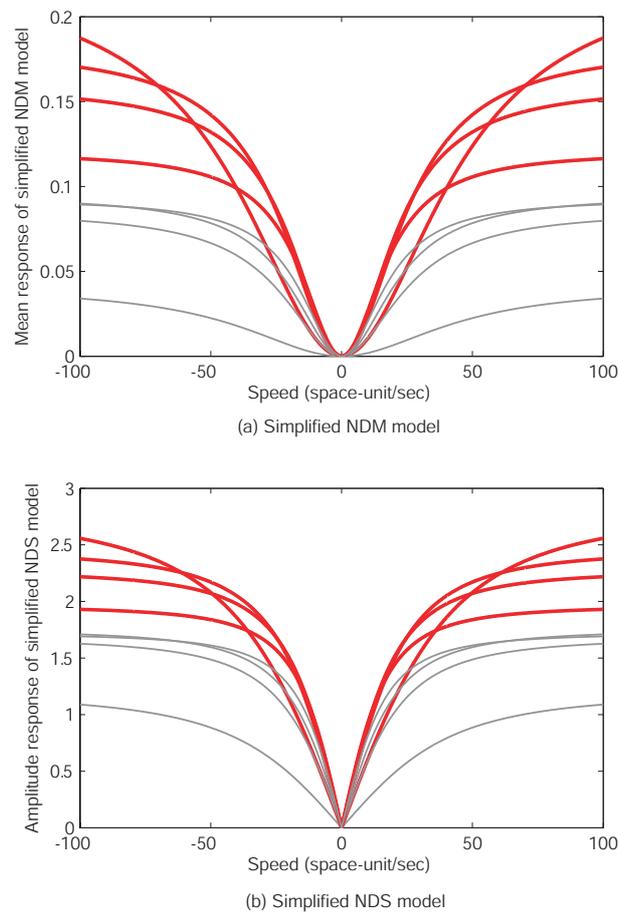


FIGURE 5.9. Comparison of the response of the simplified non-directional models to flicker and motion. (a) The flicker response of the simplified NDM model is represented by thin lines while the response to motion is shown by the thick traces. Multiple curves are responses at spatial frequencies 0.17, 0.14, 0.12, and 0.07 cycles per space-unit. The ratio of the mean response for motion and flicker is the same as that for the original NDM model. (b) The simplified NDS model's response to flicker and motion stimuli at the same spatial frequencies as above. Again, the thin curves represent flicker responses and the thick curves represent motion responses. The ratio of the mean response for motion and flicker is the same as that for the original NDS model.

CHAPTER 6

ANALOG VLSI IMPLEMENTATION OF SPEED ESTIMATION SENSORS

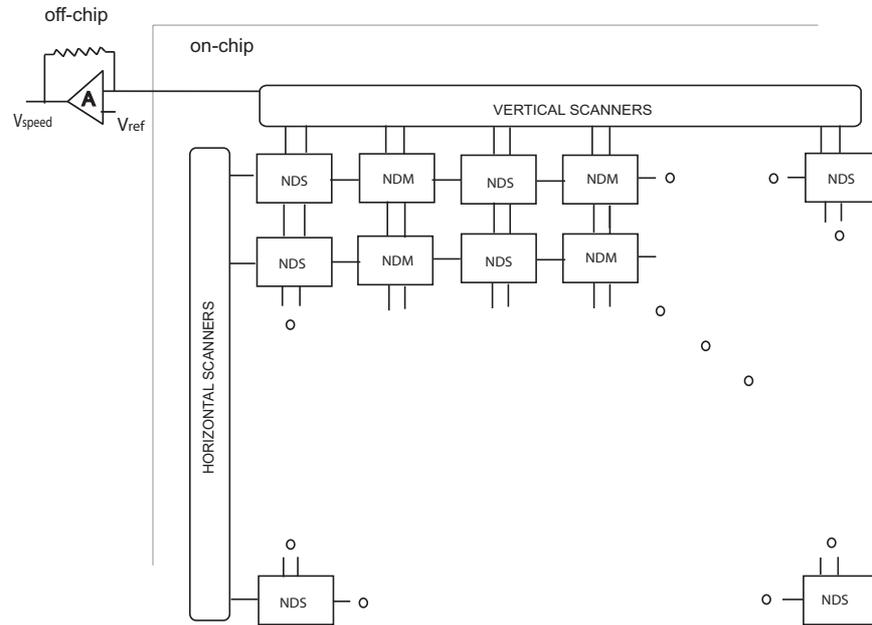
The non-directional models described in the previous chapter may be implemented in hardware by utilizing a conventional architecture for visual navigation. Such a conventional system usually employs an analog visual sensor communicating with a digital processor through an analog-to-digital converter. For a high resolution visual sensor operating on a limited power budget, the cost of high-speed data transfer and processing time is often a bottleneck. The insect visual system solves this problem in an elegant fashion. It houses both the sensor (photoreceptor) and the processing elements (neurons) in close proximity and processes the visual information in parallel. Also, the computation on the transduced visual signal is performed in the analog domain and with very little energy cost. This motivates the engineering of analog VLSI chips that have both the data acquisition sensor and the processing circuitry on the same die. Such neuromorphic architectures utilize an array of sensor circuits to transduce and parallel process the visual information in the image focal plane itself. This bypasses the problems of data communication and analog-to-digital conversion and makes the neuromorphic system faster and more power efficient. Also, since these systems utilize analog circuits to process the data, their output may be used to directly create a navigation control voltage for an autonomous robot. Neurobiological algorithms have been successfully implemented for the navigation of autonomous robots and in the design of visual sensors (Horiuchi *et al.*, 1992; Kramer *et al.*, 1995; Higgins and Pant, 2004a; Harrison, 2005).

In this chapter, we present the design of an analog VLSI speed sensor that may be used for obstacle avoidance by estimating relative depth of objects in a scene. This chip implements the non-directional models discussed in the previous chapter. An analysis of the circuit components and the characterization results from the chip are also presented. The initial results from this chip have been reported previously in Pant and Higgins (2007).

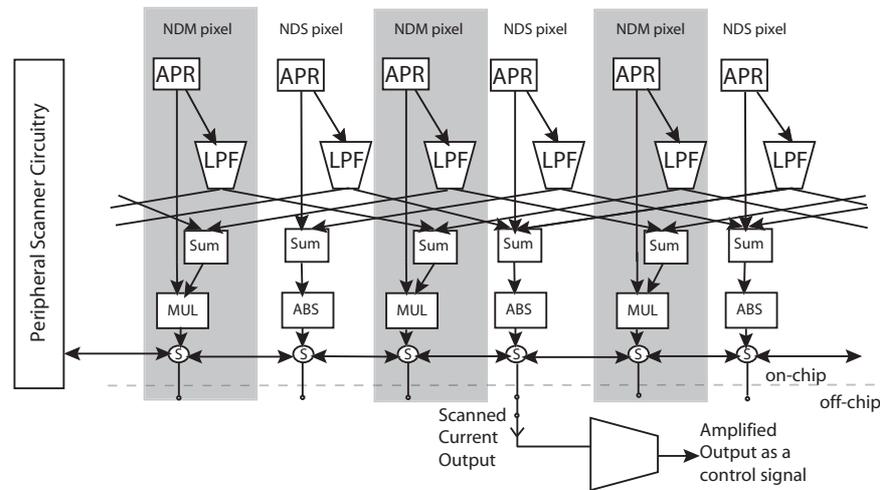
6.1 Circuit architecture

A chip incorporating both the non-directional multiplication (NDM) and non-directional summation (NDS) units discussed in Chapter 5 was fabricated in a $0.18\ \mu\text{m}$ CMOS process through MOSIS on a $4\times 4\ \text{mm}^2$ die. The chip implements the original NDS and NDM models which have low-pass filter units. However, since the low-pass filter circuits on the chip can be turned off by electronically changing a bias voltage, we were also able to implement the simplified NDM and NDS models. The two pixel grids were interleaved with each other as shown in Figure 6.1a. The size of each grid was 70×70 , and the total size of the array was 140×140 .

Each pixel on the chip computes a nondirectional motion output with respect to its adjacent two pixels of the same type (see Figure 6.1b). An adaptive photoreceptor circuit (APR) transduces light incident on a pixel into an electrical signal. This circuit can operate over five orders of magnitude of light intensity and has bandpass temporal frequency characteristics (Delbrück and Mead, 1993). A g_m -C type first-order low-pass filter (LPF) with tunable time constant implements the next computational block. In the case of an NDS pixel, the response of two adjacent LPF circuits are summed with the output of the central photoreceptor circuit. The computation of the amplitude response was performed by a current-mode absolute value circuit (ABS) in each NDS pixel. The NDM pixel takes the sum of two adjacent LPF circuits and multiplies it with the output of the central photoreceptor circuit. To implement the simplified NDM and NDS pixels, the LPF circuit could simply be turned off so that it passes the signal unchanged. A single-transistor gate (S) in each



(a)



(b)

FIGURE 6.1. Block diagrams of the chip architecture. (a) Floor plan of the chip showing the NDS and NDM pixels arranged in an interleaved fashion. The vertical and horizontal scanners allow for the selection of a pixel(s) by external digital inputs. The output from the chip is a current that is processed by a current sense amplifier to generate a voltage signal. (b) Internal blocks within the NDS (white background) and NDM pixels (gray background). An adaptive photoreceptor (APR) transduces the incident light intensity into an electrical signal. The output of this circuit has bandpass temporal frequency characteristics and therefore no explicit high-pass filter circuit was required on the chip. In the NDS pixel, this signal is then summed with the low-pass filter (LPF) output of the two adjacent NDS pixels. The LPF units in all the pixels may be turned off to implement the simplified NDS and NDM pixels. This signal is further processed by an absolute value circuit (ABS) in the NDS pixels. In the case of an NDM pixel, the LPF outputs from the two adjacent NDM pixels are added and then multiplied with the output of the central APR. A single transistor switch (S) is used to route the output from a pixel to the external current sense amplifier. The peripheral scanner circuitry regulates the opening and closing of the switches (S) based on external digital inputs.

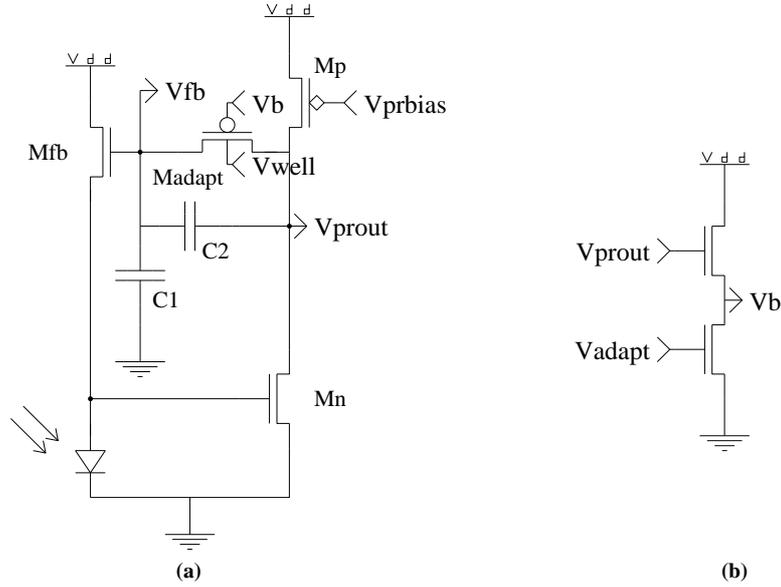


FIGURE 6.2. Adaptive Photoreceptor circuit. (a) Voltage fluctuations caused by the photodiode at the gate of the transistor M_n are amplified by the Early voltage amplifier (M_n and M_p). This signal is fed back to the photodiode node by a source follower transistor M_{fb} . The feedback network has a faster capacitive-divider signal path (via C_1 and C_2), and a slower resistive path (via M_{adapt}) with a time constant controlled by a bias voltage V_b . (b) This two-transistor circuit generates the bias voltage V_b for the adaptive photoreceptor circuit.

pixel routes output from the pixel to an external current sense amplifier circuit, and is controlled by on-chip digital scanner circuitry. A combination of these switches may be activated simultaneously to generate the net image-speed output from the non-directional models by current summing.

The details of the subcircuits used to implement different computational blocks is described below.

6.1.1 Adaptive photoreceptor

An adaptive photoreceptor circuit (Delbrück and Mead, 1993) was used for phototransduction (see Figure 6.2). This circuit converts the intensity pattern incident on its photodiode into an electrical signal, and uses a feedback mechanism to operate reliably over five orders of magnitude of illumination change. In our implementation, two outputs from this circuit were utilized: a logarithmically encoded contrast voltage signal V_{prout} , and its long-time mean response V_{fb} . This circuit has a temporal bandpass frequency characteristic, and therefore implicitly performs the function of the high-pass filter stage in the non-directional models. In a $0.18 \mu\text{m}$ process, the active layer (the topmost doped layer in a silicon wafer that is used as the source and drain terminals of a MOSFET) is very shallow and provides less volume for photon absorption than a process with larger feature size. The well layer in this process, on the other hand, is deeper than the active layer and hence provides more volume for photon absorption. Therefore, to maximize the photosensitivity of the circuit, the photodiode was implemented as a well-type p-n junction. The *fill factor* (percentage of pixel area devoted to phototransduction) was 13%-17% of the pixel layout based on the pixel type as will be explained in Section 6.2. The photodiode was surrounded by a guard ring to prevent minority carriers from diffusing towards other transistor elements in the pixel.

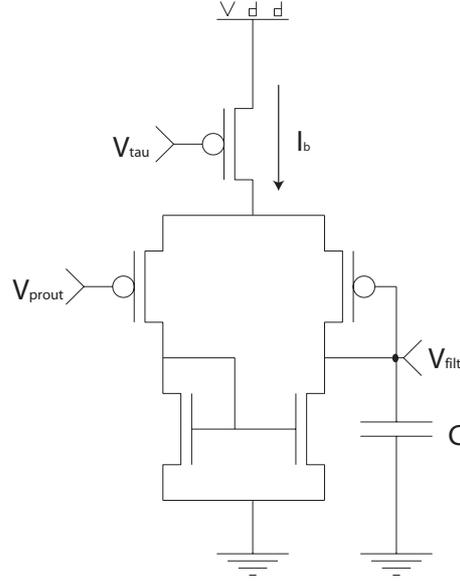


FIGURE 6.3. A g_m -C low-pass filter circuit. A p-type transconductance amplifier has its output node V_{filt} connected to its negative input and a capacitor C . The bias voltage V_{tau} controls the bias current I_b through the circuit.

6.1.2 Low-pass filter

The first-order low-pass filter unit utilized in the NDS and NDM models was implemented as a five-transistor transconductance amplifier based g_m -C filter. A PMOS-type g_m -C filter is shown in Figure 6.3. A bias current I_b was used to control the time constant τ of this first-order filter circuit as illustrated by the frequency domain analysis in the following equation:

$$V_{filt}(s) = V_{prout}(s) \cdot \frac{1}{1 + \tau \cdot s} \quad (6.1)$$

where

$$\tau = \frac{C}{g_m} = \frac{C \cdot U_T}{\kappa \cdot (I_b/2)} \quad (6.2)$$

In the above equation, U_T is the thermal voltage, and κ is the back-gate coefficient. All transistors were operated in the subthreshold regime of the MOSFET. The bias current of this circuit was used to set the time constant of the filter, and this current may in turn be controlled by an external bias voltage V_{tau} :

$$I_b = I_0 \exp\left(\frac{\kappa \cdot (V_{dd} - V_{tau})}{U_T}\right) \quad (6.3)$$

where V_{dd} is the power-supply voltage, and I_0 is the current when $V_{tau} = V_{dd}$.

The NDM pixel used a PMOS-type circuit as described above, while the NDS pixel utilized an NMOS-type circuit where all the NMOS transistors were replaced by the PMOS transistors and vice versa. This ensured that the DC voltage level of the respective pixels were in the proper operating range of the next circuit stage.

This low-pass filter stage may be turned off by setting the time constant of the filter to a small value. This is done by increasing the gate-to-source voltage of the bias transistor such that I_b increases. A large value of I_b makes the time constant τ small such that the cut-off frequency of the

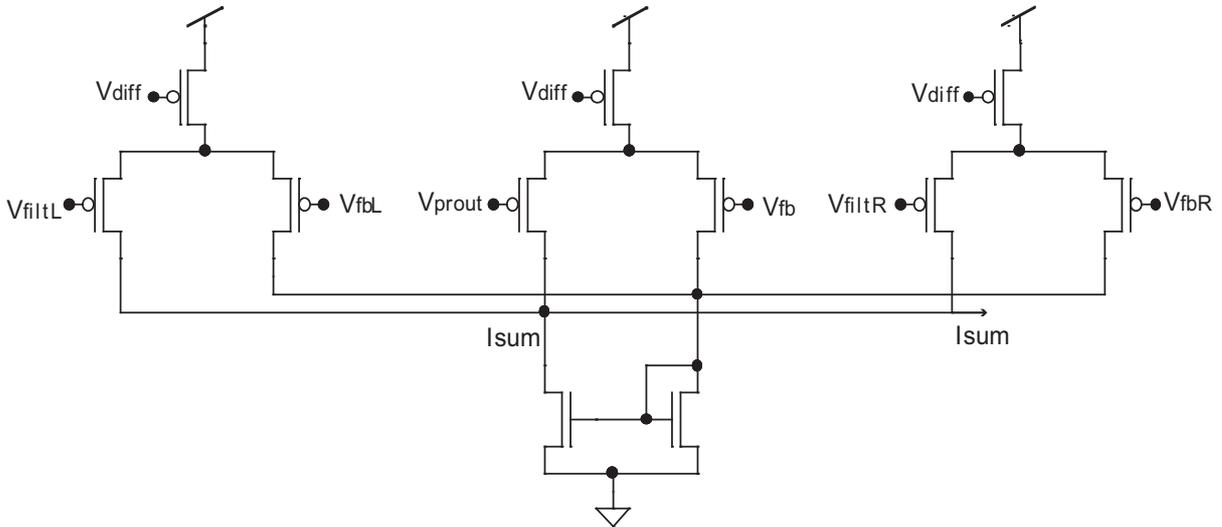


FIGURE 6.4. A current-mode differential voltage addition circuit. V_{diff} is the bias voltage that provides the current in the three branches of the circuit. I_{sum} represents the sum of the three currents induced by the difference of the low-pass filter and photoreceptor voltages and their respective mean values.

circuit becomes high. Thus the circuit allows both low and high frequencies to pass and the input and output voltages are almost equal.

6.1.3 Current-mode differential voltage addition circuit

In NDS pixel, the output of the low-pass filters from adjacent pixels are summed with the central photoreceptor output. This was achieved by utilizing a current-mode differential voltage addition circuit as shown in Figure 6.4. At the first stage of this circuit, the voltage signals of the low-pass filters from the two neighboring pixels V_{filtL} and V_{filtR} , and the photoreceptor output V_{prout} from the same pixel were subtracted from their mean values (V_{fbL} , V_{fbR} , and V_{fb} , respectively) obtained from the long-term mean response of the respective adaptive photoreceptor circuits. These differences were converted into current signals and summed together by utilizing Kirchoff's Current Law (KCL). The output of this circuit is a bidirectional current I_{sum} .

6.1.4 Absolute value circuit

The NDM pixel encodes stimulus speed in its mean output which can be easily computed off-chip. The NDS pixel, on the other hand, encodes speed in the amplitude of its output which is easier to compute on-chip. The amplitude of a signal is proportional to the mean of its absolute value. We use this idea to extract the amplitude information from the NDS pixel output. The current output of the addition circuit is fed into an absolute value circuit (see Figure 6.5). The absolute value circuit provides a source or a sink for the current going in or out of the addition circuit. The current mirror acts as a sink while the cascode transistor sources any current required by the addition circuit. The final output is a bidirectional current I_{out} . Off-chip, we compute the mean of this signal which is proportional to the amplitude of the response of the NDS pixel.

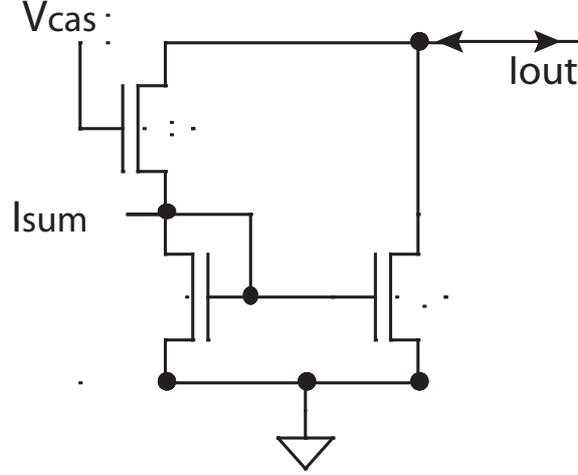


FIGURE 6.5. A current-mode absolute value circuit (Toumazou *et al.*, 1990). A bias voltage V_{cas} controls a cascode transistor that isolates the voltage fluctuation at the input node from the output. The output is a bidirectional current I_{out} .

6.1.5 Differential add-multiply circuit

The addition and multiplication stages of the NDM model were implemented together in a novel differential mode add-multiply circuit that we designed for this chip based on the Gilbert multiplication cell (Gilbert, 1974). The Gilbert cell was modified to perform both the addition and the multiplication operations. A four-quadrant differential add-multiply circuit is shown in Figure 6.6. The function implemented by this circuit is $(A + B) \times C$. A differential input scheme was adopted in order to maximize the dynamic range of operation. The difference between each input and its mean was converted into a current. The inputs to this circuit came from the low-pass filter circuits in the two adjacent NDM pixels, and the output of the adaptive photoreceptor circuit. The mean response of the inputs were connected to the long-term mean output of the respective photoreceptor circuits. The removal of the mean from the input makes the output of this circuit resistant to any DC perturbation of its inputs. This effectively counters the process-variation induced differences in the operating level of the transistors across the chip. We also incorporated an adjustable well potential (V_w) for the lower four p-type transistors (M7-M10) to enable them to operate at the same input voltage levels as the upper input transistors (M2, M3, M5, M6), as first reported by Harrison (2005). The well terminals of all other p-type transistors were connected to the chip's most positive potential (V_{dd}). All the transistors in this circuit operate in the subthreshold regime of the MOSFET. The circuit may be analyzed to get the following expression for the output current:

$$I_{out} = I_b \cdot \tanh\left(\kappa \frac{V_c - V_{cr}}{2U_T}\right) \times \left[\tanh\left(\kappa \frac{V_a - V_{ar}}{2U_T}\right) + \tanh\left(\kappa \frac{V_b - V_{br}}{2U_T}\right) \right] \quad (6.4)$$

where I_b is the bias current, κ is the back-gate coefficient, U_T is the thermal voltage, and V_{ar} , V_{br} , and V_{cr} are the mean values about which the inputs V_a , V_b , and V_c vary. Since the hyperbolic tangent function is linear for small values, we can approximate the above expression for small input voltages as:

$$I_{out} \approx I_b \left(\frac{\kappa^2}{4U_T^2} \right) \cdot (V_c - V_{cr}) \times [(V_a - V_{ar}) + (V_b - V_{br})] \quad (6.5)$$

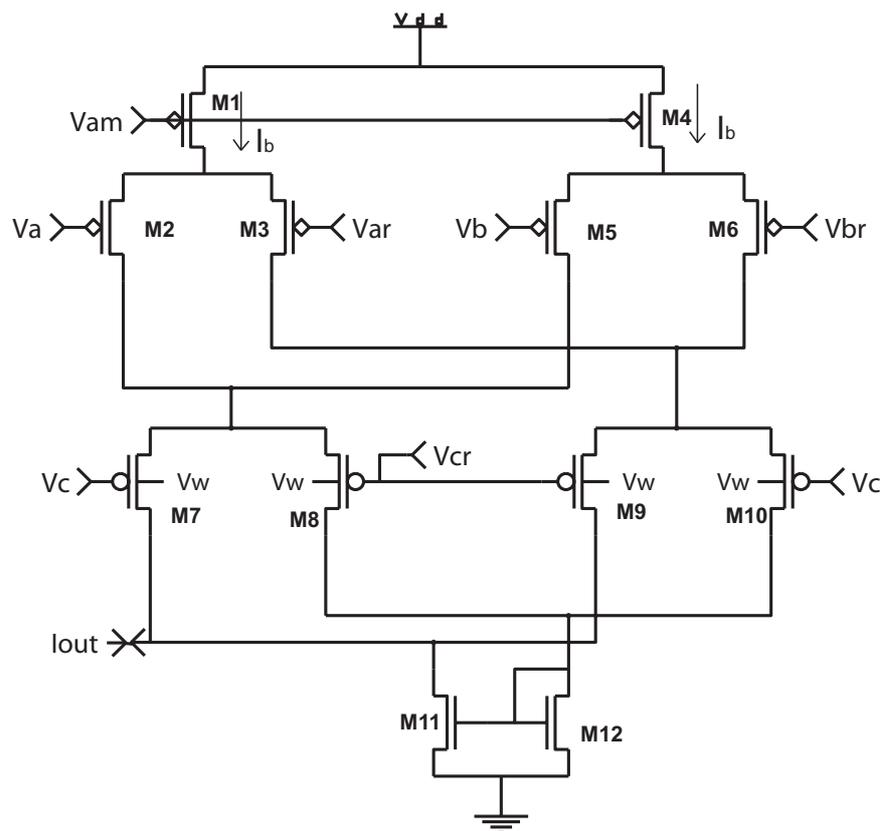


FIGURE 6.6. A differential add-multiply circuit. Inputs V_a and V_b are added and then multiplied with the input V_c . V_{ar} , V_{br} , and V_{cr} are the mean values about which these inputs vary. The multiplier input transistors M7, M8, M9, and M10 have their well terminals controlled by a bias V_w . All other PMOS well terminals were connected to V_{dd} . A bias voltage V_{am} controls the currents I_b flowing through this circuit. The output of the circuit is a bi-directional current I_{out} .

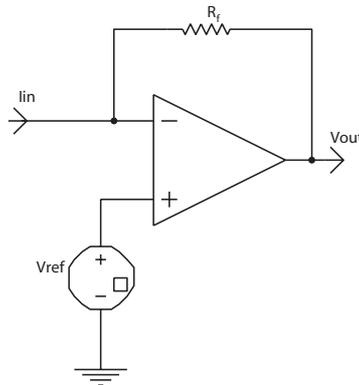


FIGURE 6.7. A current sense amplifier circuit. I_{in} is the input current, V_{ref} is the reference voltage, and V_{out} is the output voltage. A feedback resistor R_f sets the gain of the amplifier.

As can be seen from the above equation, the circuit computes the desired function over a small linear range of its inputs. The output saturates when the inputs exceed this linear range (experimentally determined to be about 250 mV: see Section 6.3.1).

6.1.6 Scanner circuit

The peripheral digital scanner circuitry was designed on the chip to access the pixel core. Scanners are addressable flip-flop circuits and act as an indexing scheme to route a signal to and from a specific pixel (addressed by its x- and y-position) to a bonding pad on the chip (Mead and Delbrück, 1991). A single-transistor MOS switch was used in every pixel and could be digitally switched on to route the output of that specific pixel to an output pad on the chip. In most operating conditions, a combination of outputs from several pixels was computed by KCL sum of the current outputs to estimate the net image speed of a stimulus.

6.1.7 Current sense amplifier

A current sense amplifier circuit was implemented external to the chip to convert the tiny current output from the pixels into a measurable voltage signal. Figure 6.7 shows the current sense amplifier configuration that we utilized on our circuit board. An operational amplifier was connected in a negative feedback configuration with a feedback resistor R_f . The output pin of the chip was connected to the negative input of the operational amplifier and provided the input current I_{in} . The positive input of the amplifier was connected to a reference voltage V_{ref} . For this circuit, the output voltage may then be computed to be

$$V_{out} = V_{ref} + I_{in} \cdot R_f \quad (6.6)$$

From the above equation we can conclude that the current sense amplifier output voltage has an offset equal to the reference voltage V_{ref} , and a gain equal to the feedback resistor R_f .

6.2 Chip layout

The layout of the circuits was drawn using L-Edit (Tanner Research, Inc.). The chip was designed to have an array of 140×140 interleaved NDS and NDM type pixels. Two different layouts were fabricated for each pixel type with the same outer dimensions, but differing in fill factor. One pixel type was designed with a high fill factor and smaller capacitors (pixel type 1), and the other was

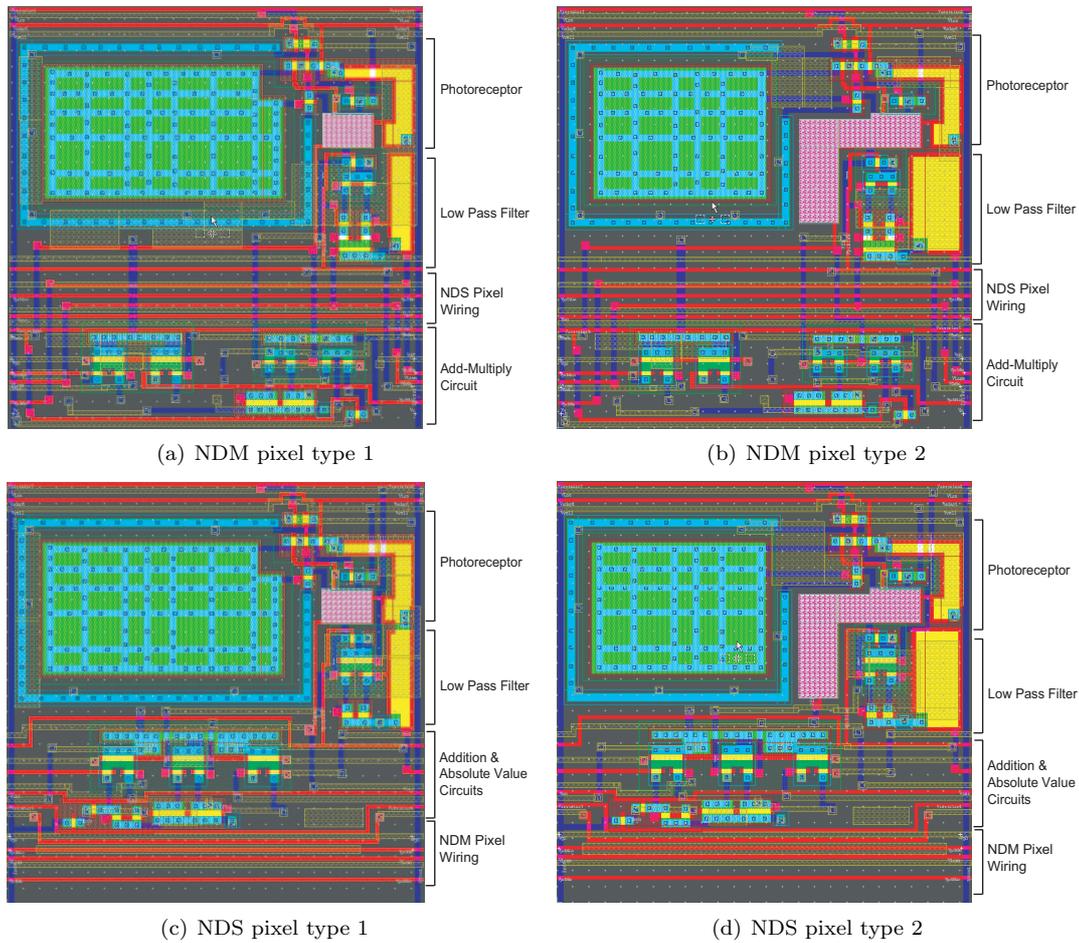


FIGURE 6.8. Layout of the NDM and NDS pixels showing the different sub-circuits.

designed with bigger capacitors for better control over the circuit time constants but lower fill factor (pixel type 2). Figure 6.8 shows the layout of the NDM and NDS pixels. 70×70 pixels of each type were fabricated on a single die in a $0.18 \mu\text{m}$ CMOS process. The type 1 NDS and NDM pixels had a fill factor of 17% and this led to their having a much smaller low-pass filter capacitor. The type 2 pixel had a fill factor of 13% with a bigger low-pass filter capacitor as seen in Figure 6.8. The NDS and NDM pixels were juxtaposed as shown in Figure 6.9 to form a grid. The time constant of the low-pass filter did not affect the output of the simplified model, and the outputs were virtually identical for the two pixel types with the low-pass filter circuit turned off. The layout of the whole chip with the peripheral scanner circuitry and the bonding pads is shown in Figure 6.10a. A photomicrograph of the fabricated integrated circuit is shown in Figure 6.10b.

6.3 Characterization

The fabricated sensor was tested using the setup shown in Figure 6.11. The chip was mounted on a custom designed circuit board with bias and interface circuitry. Two data transfer ports, one analog and the other digital, were placed on the board (see Figure 6.12). The digital port was used to interface the circuit board with a computer that provided control signals for the peripheral scanner

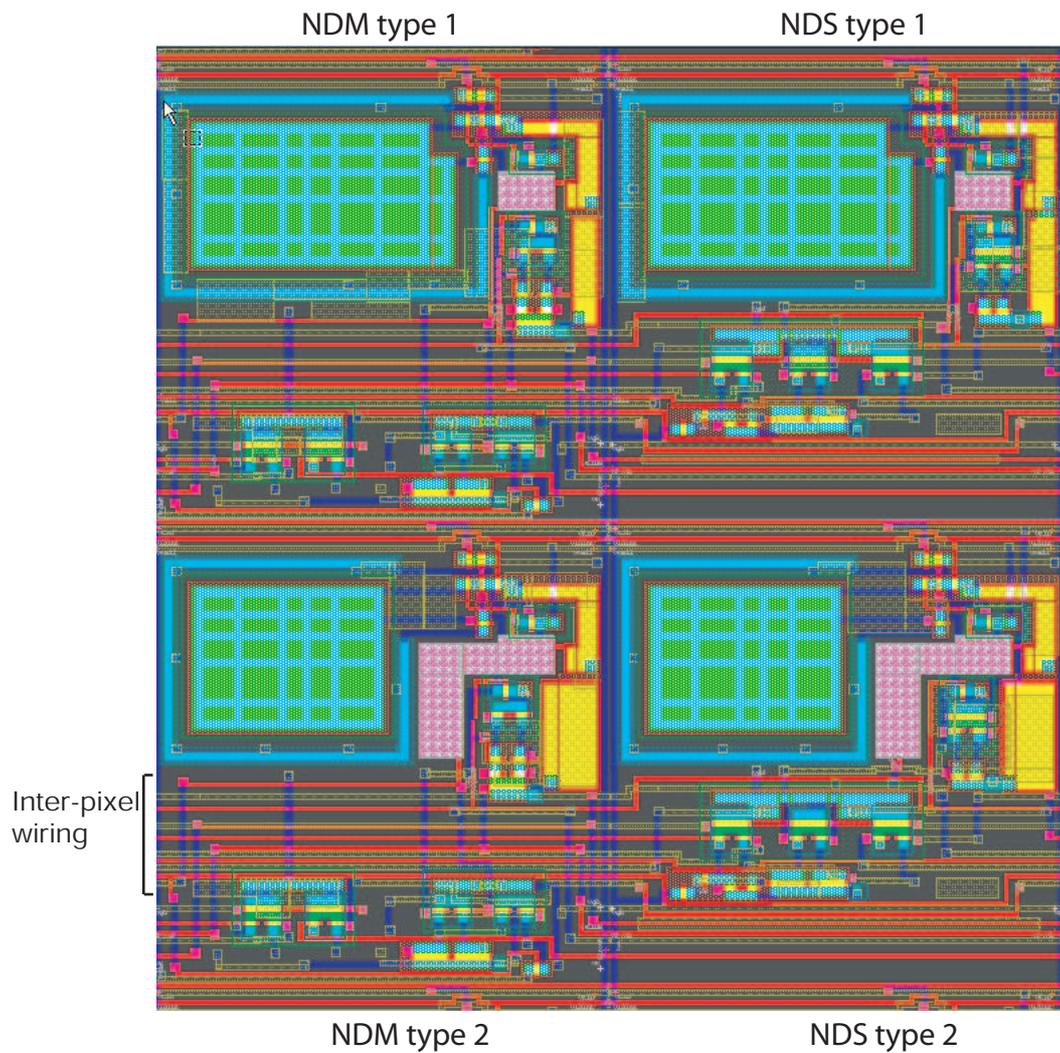


FIGURE 6.9. Layout of the juxtaposed NDS and NDM pixels showing the inter-pixel wiring in the center of the chip. All the rows above these central rows comprised type 1 pixels, while all the rows below them were type 2 pixels.

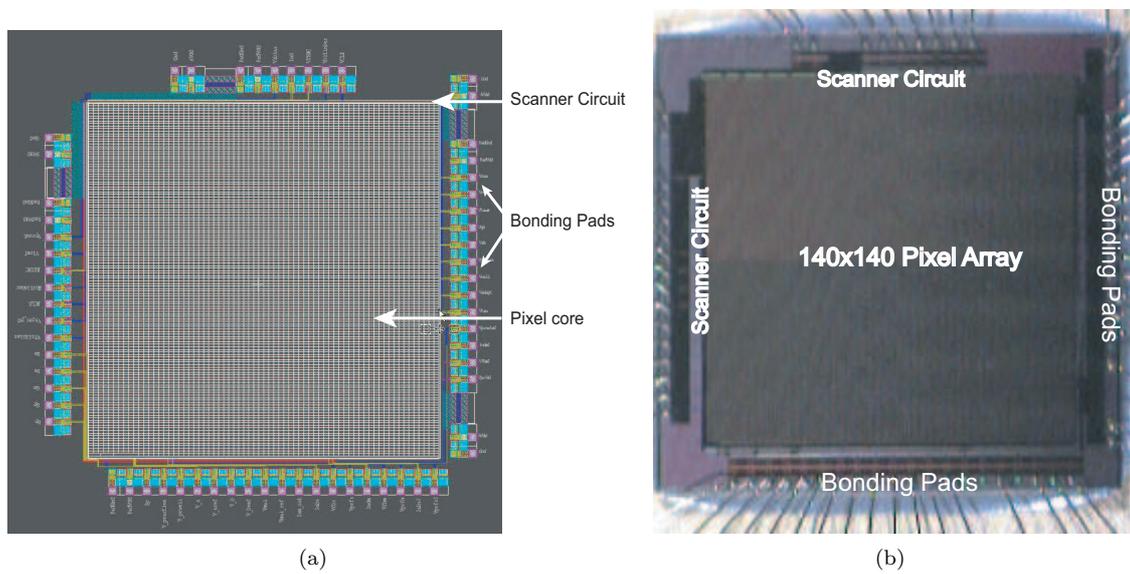


FIGURE 6.10. The ND chip. (a) Layout. (b) Photomicrograph of the integrated circuit.

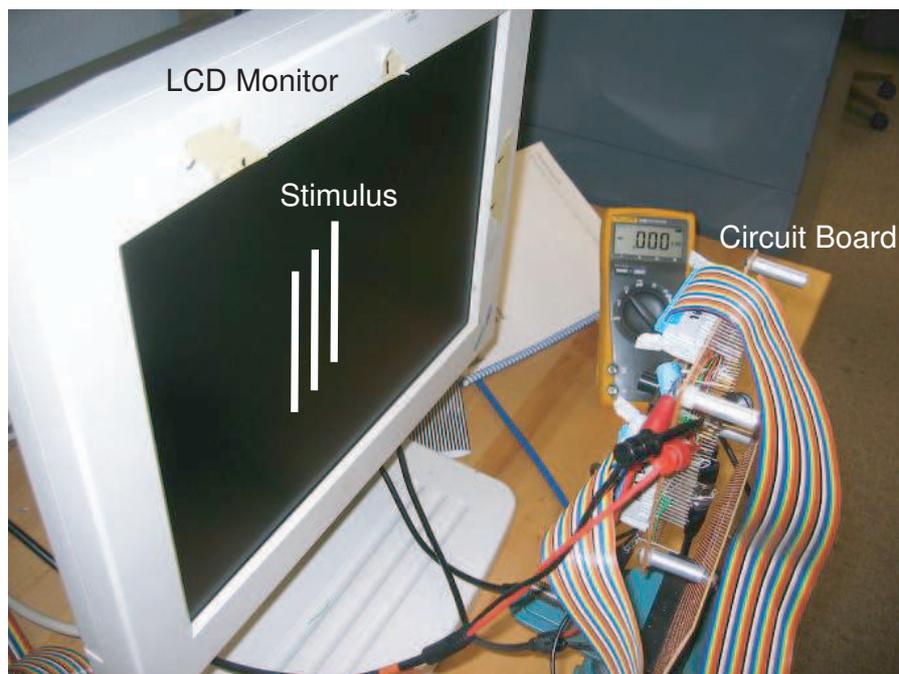


FIGURE 6.11. Chip characterization setup. The circuit board was aligned so as to view the image on an LCD monitor. The monitor display was controlled by a C program on a computer. The computer also controlled inputs to the scanner circuitry on the chip, and recorded analog voltage outputs from the current sense amplifier.

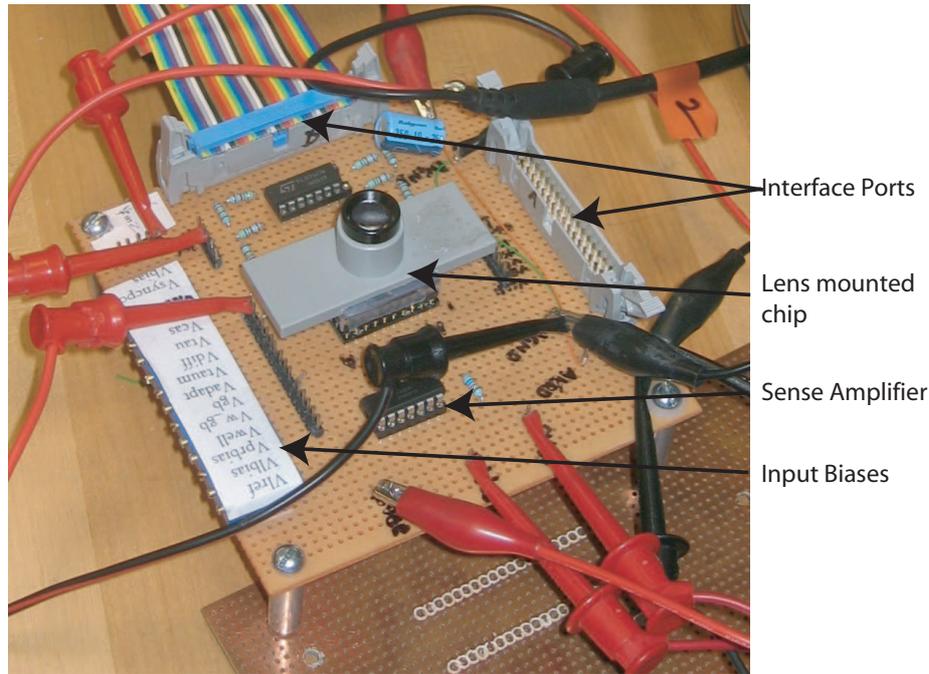


FIGURE 6.12. Circuit board for testing the sensor showing the interface ports, bias circuits, and the lens-mounted chip.

circuitry. The analog port was used to record the data from the chip. The output from the chip, a current signal representing the sum of NDS or NDM pixels, was converted into a voltage signal by a current sense amplifier with a gain of 8.7×10^5 V/A. The stimulus was generated using a computer program and displayed on an LCD monitor. The chip was mounted with a lens to focus the image of the stimulus on the sensor, and was held such that the die was aligned parallel to the monitor. The chip was stationed at a distance of 16.4 cm from the monitor. The focal length of the lens mounted on the chip was 13.3 mm.

The add-multiply circuit was characterized separately from the rest of the pixels in the circuit. This was performed by providing direct voltage inputs to the different terminals of a test circuit fabricated on the chip, and the current output was again processed into a voltage signal by the current sense amplifier on the circuit board. The characterization results of this add-multiply circuit, and the NDM and NDS pixels, are presented below.

6.3.1 Differential add-multiply circuit

In the case of the differential add-multiply circuit, the power supply for the test circuit was set at 2.2 V and three different inputs were provided using stable external power supplies. The differential add-multiply circuit performs the following operation: $(V_c - V_{cr}) \times ((V_a - V_{ar}) + (V_b - V_{br}))$, where the notation is the same as in Section 6.1.5. From here on, we call these differential voltages V_{cd} , V_{ad} , and V_{bd} , respectively. We tested the linearity of the circuit for several cases. First, we tested the linearity while the multiplier voltage V_{cd} was varied. To test this, we fixed the input voltages V_{ad} and V_{bd} at 42 mV and 32 mV respectively, and varied V_{cd} from -206 mV to 194 mV. As seen in Figure 6.13a, the output voltage for V_{cd} between -106 mV and 94 mV is roughly linear. The variance of the measured output is within 16% variance of a linear fit shown by the dark line in the figure.

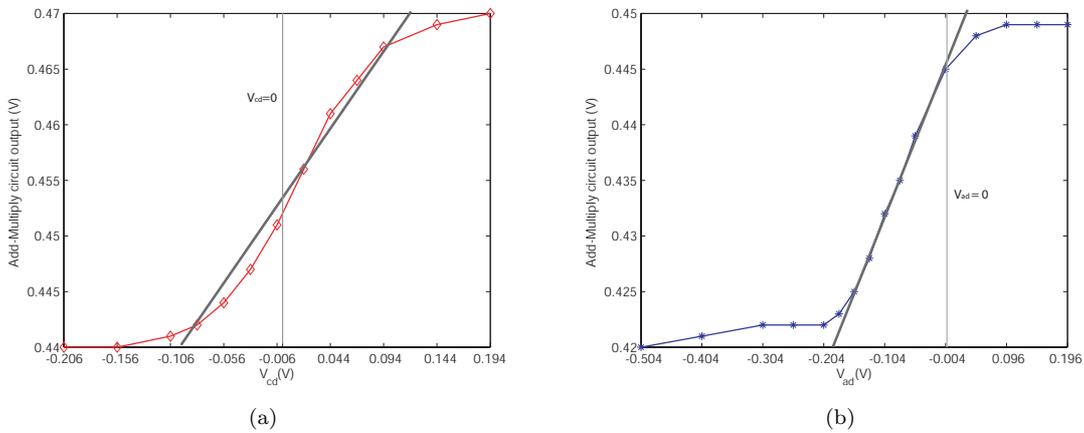


FIGURE 6.13. Response of the add-multiply circuit as V_{cd} (a) and V_{ad} (b) are varied. The dark line shows a linear fit to the data. The gray vertical line indicates when the input differential voltages V_{cd} and V_{ad} are zero in plots (a) and (b) respectively.

The linear range of V_{cd} is approximately 200 mV after which the output saturates as expected from Equation 6.4.

In the next case, one of the multiplicand voltages V_{ad} was varied. We fixed V_{bd} and V_{cd} at 35 mV and 24 mV respectively. V_{ad} was varied over a range between -504 mV and 196 mV. The output of the circuit was linear over a range of roughly 200 mV as seen by the linear trend line in Figure 6.13b. Outside this range, the output of the circuit saturates. The output is shifted towards the left side of the zero value of V_{ad} due to the positive value of V_{bd} which gets added to V_{ad} .

We next varied both V_{ad} and V_{bd} while keeping V_{cd} fixed at -76 mV. V_{bd} was varied from -285 mV to 315 mV, while V_{ad} was chosen to be either -204 mV, -81 mV, 43 mV, or 121 mV. In this case, as V_{bd} is varied and the value of V_{ad} is increased, the output of the circuit should have different DC offsets but the same slope. As shown in Figure 6.14, the output of the circuit increases linearly with the sum of V_{ad} and V_{bd} (individual traces in the figure). Since V_{cd} is negative here, the output of the circuit decreases with an increase in V_{bd} . Similarly, as V_{ad} decreases from 121 mV to -204 mV in four steps (four traces in the figure), the DC offset of the output increases. The saturation of the response with increasing V_{ad} is seen by the overlapping of the two lower traces in the figure.

The characterization results of the entire circuit are shown in Figure 6.15. We fixed V_{ad} at 43 mV, and then independently varied either V_{bd} or V_{cd} to obtain the plots shown in the figure. Note that the output saturates as V_{bd} exceeds a certain voltage range (nearly 300 mV). Also, there is a saturation with V_{cd} as can be seen by the banding of traces at the top and bottom of the figure (similar to the overlapping traces in Figure 6.14). The linear range for V_c was nearly 400 mV. The trend in the linear region matches closely with the theoretical response of the function $(A + B) \times C$. The positive voltage level of V_{ad} makes the responses on the left half of the figure saturate at higher (lower) values, based on the negative (positive) voltage level of V_{cd} . This happens because for positive V_{ad} , the sum of the inputs V_{ad} and V_{bd} becomes less negative when V_{bd} is negative and more positive when V_{bd} is positive. Since the linear range of the circuit is constant at about 300 mV, this causes the linear range of the output to be smaller on the right half of the figure.

The saturation of the response beyond the linear range is a useful feature of this circuit. In biological systems, we commonly observe a similar trend where the neuronal circuits saturate and thereby impose an upper or lower limit on the system. Likewise, in robotic applications, it is often desired for system stability reasons that the circuits impose a hard limit on the control signal. The

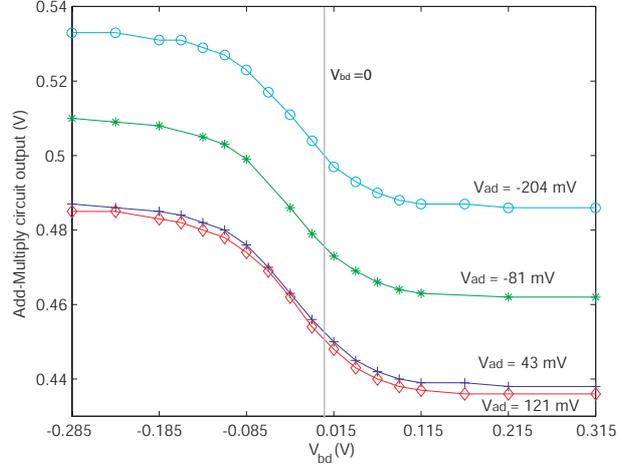


FIGURE 6.14. Response of the add-multiply circuit versus the input voltage V_{bd} . The four traces represent data at four different values of V_{ad} (-204 mV, -81 mV, 43 mV, and 121 mV). The value of V_{cd} was -76 mV. The zero level of V_{bd} is shown by the gray vertical line.

saturation of the output in this circuit provides exactly that.

6.3.2 NDM pixel characterization

In this section, we present the results from the simplified NDM pixels. As described in Section 6.1.6, the scanner circuit can be provided with an input to select either the NDS or the NDM pixels. The low-pass filter circuit was turned off to emulate the simplified non-directional models. We activated eight spatially random NDM pixels in a single row, and recorded their response using a current sense amplifier. A random selection of pixels ensures that the input phase of the visual stimulus does not systematically affect the output of the pixels. The voltage output from the current sense amplifier had an offset due to the summation of the pixel currents. Figure 6.16a shows the raw data from the chip when presented with a sinusoidal grating of spatial frequency 0.13 cycles/pixel. The speed of the stimulus was stepped from 8 to 26 pixels/second with each pixel being $23.175 \mu\text{m}$ wide. The time trace in the figure suggests that the response of the sensor increases linearly with stimulus speed.

We further characterized the NDM sensor by plotting its response against the speed of the stimulus at various spatial frequencies. The speed of the stimulus ($v = \omega_t / \omega_x$) was varied from -30 to +30 pixels/second. The spatial frequency was varied from 0.14 cycles/pixel to 0.2225 cycles/pixel in steps of 0.0275 cycles/pixel. Data from the chip was collected as a sum of eight randomly selected pixels in a single row on the chip over five repetitions of the visual stimulus. The data from these repetitions were processed to compute mean and standard deviation of the response at all speeds for every spatial frequency. The average speed response from the NDM pixels are plotted in Figure 6.16b. The response of the chip at low speeds (between ± 6 pixels/second) was found to be dependent on the mean luminance of the screen and not the stimulus speed due to light-induced leakage. At these speeds, the temporal frequency of the stimulus was very small and the grating drifted very slowly on the screen. The solid line in the figure represents the linear response with speed. It can be seen that the response of the NDM pixels is relatively independent of the spatial frequency.

We also plotted the average response of the NDM pixels while the spatial frequency and the

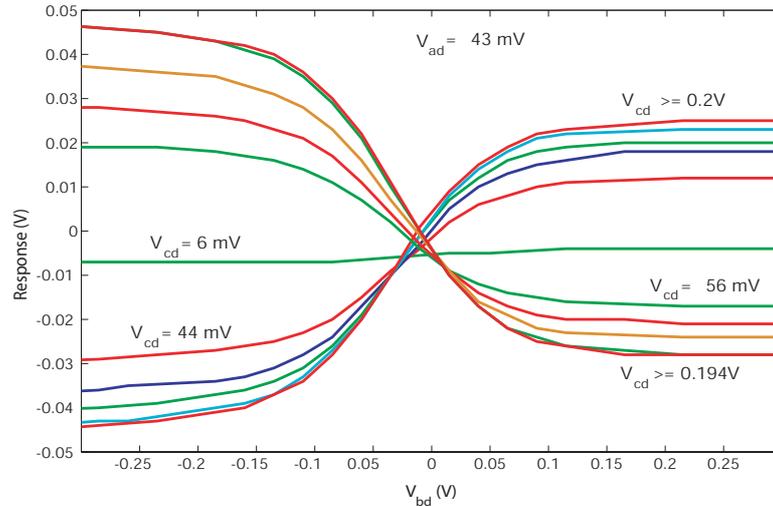


FIGURE 6.15. Output characteristics of the differential add-multiply circuit. The current-mode output of the circuit was converted into a voltage signal using a current sense amplifier with a gain of 8.7×10^5 V/A. Input V_{ad} was fixed at 43 mV, while inputs V_{bd} and V_{cd} were varied. The output shows a linear region that matches with the functional response $((A + B) \times C)$, and a non-linear region where the output saturates (see text for details).

speed of the stimulus were varied. Data were recorded from the same set of pixels as in the previous experiment. The speed of the stimulus was varied from -30 to 30 pixels/second. The spatial frequency was varied from 0.08 to 0.225 cycles/pixel. Figure 6.16c shows a contour plot of the mean response versus the spatial frequency and speed of the stimulus. A dark shade represents a lower response magnitude while a lighter shade indicates a higher magnitude. We have superimposed contour lines (white traces in the figure) of the response on this graph. The plot shows similar trends as were seen in the contour plots of the simplified NDM model in Chapter 5. However, at low speeds (less than ± 6 pixels/second), the data shows that the response of the chip is higher for low spatial frequencies (less than 0.14 cycles/pixel) as compared to the response at higher spatial frequencies (white island like contour lines in the middle of the graph). At these spatial frequencies, as discussed in the case of the speed curve in Figure 6.16b, the temporal frequency is very low (a speed of 10 pixels/second at 0.1 cycles/pixel requires a temporal frequency of 1 Hz). For slower speeds, the temporal frequency is lower than 1 Hz. At such low temporal frequencies, the sensor is not able to detect a change in the speed of the grating and rather responds to the mean luminance of the screen due to light-induced leakage.

In Figure 6.16c, for lower spatial frequencies we can see the saturation of the response at moderate (20 pixels/second) to high (30 pixels/second) speeds as seen by the curving of the contour lines parallel to the speed axis. In the intermediate spatial frequency range (0.12 to 0.225 cycles/pixel), the response magnitude does not change much with spatial frequency (parallel to the y-axis) but changes almost linearly with speed. We could not probe the chip at spatial frequencies higher than 0.225 cycles/pixel due to resolution limitations of the LCD monitor.

The relative independence of the response to the spatial frequency of the stimulus may be further seen in Figure 6.16d. It shows the average response and standard deviation of the NDM pixels at speeds of 20 and 8 pixels/second (upper and lower trace, respectively). We see that at a speed of 8 pixels/second, the response is relatively flat against a range of spatial frequencies (from 0.12 to 0.225 cycles/pixel as indicated by the trend line). The standard deviation is also small in this range.

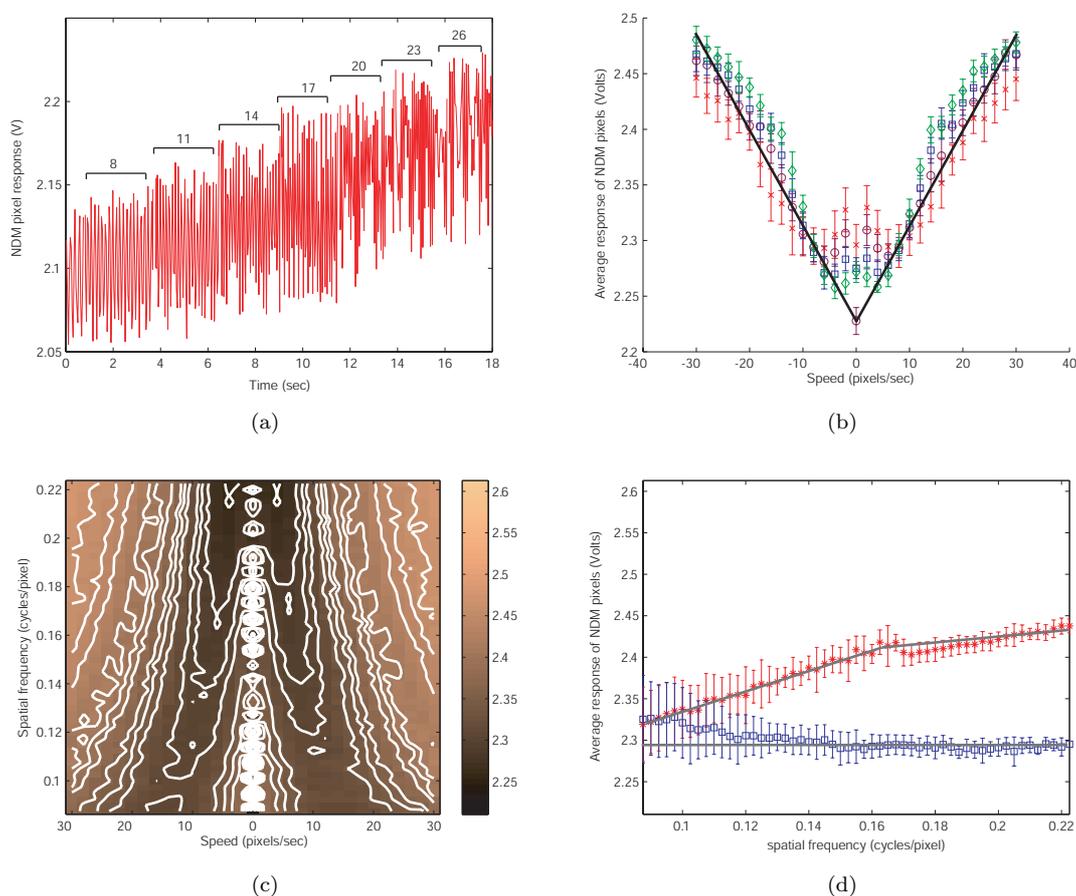


FIGURE 6.16. Response of the NDM pixels to a sinusoidal stimulus. (a) A time trace of the response as stimulus speed is stepped from 8 to 26 pixels/second as indicated by the brackets in the figure. (b) Response of the NDM pixels (mean and standard deviation) as the speed of the stimulus is varied. Multiple sets of data collected at spatial frequencies 0.14, 0.1675, 0.195, and 0.2225 cycles/pixel are overplotted to show that the response is relatively independent of spatial frequency. The solid line represents a speed-dependent linear response. (c) Contour plot of the average response as the spatial frequency and the speed of the stimulus are varied. The magnitude increases from dark to light shade. The response magnitude increases with speed and is almost independent of the spatial frequency for a range of spatial frequency as shown by the vertical portion of the contour lines. (d) Average response and standard deviation versus spatial frequency. The upper trace shows the response at a speed of 20 pixels/second, while the lower trace shows the data at a speed of 8 pixels/second. The line segments show the trend of the data for each case.

For the higher speed trace, we find that the response increases linearly with spatial frequency up to a range (from 0.08 to 0.16 cycles/pixel) and after that the increase with spatial frequency is small as indicated by the trend line in the figure. This shows that the response for higher speed is less dependent on the spatial frequency only for a small range above 0.16 cycles/pixel. The linear change at low spatial frequencies in this case points to the gradual saturation of the response (see Figure 6.16c). The standard deviation is large at very low spatial frequencies for both the traces. This was again due to the inability of the sensor to respond in a speed-dependent manner to very low temporal frequency inputs as discussed earlier.

6.3.3 NDS pixel characterization

For characterizing the simplified NDS pixels, the low-pass filter circuits were turned off while collecting the data from the chip. The scanner circuitry was utilized to select eight spatially-random NDS pixels in a single row. The output was a sum of the current outputs of the individual pixels which was then converted into a voltage signal by a current sense amplifier. Figure 6.17a shows the raw data from the chip when presented with a sinusoidal grating of spatial frequency 0.13 cycles/pixel. The speed of the stimulus was stepped from 8 to 26 pixels/second. The plot suggests a linear increase in the chip response with speed.

Figure 6.17b shows the mean and standard deviation of the response of the NDS pixels as the speed of the stimulus was varied from -30 to 30 pixels/second. The spatial frequency was varied from 0.14 cycles/pixel to 0.2225 cycles/pixel in steps of 0.0275 cycles/pixel. The data was collected in the same manner as data for the NDM pixels. As shown in Figure 6.17b, the response increases linearly with speed and is fairly independent of spatial frequency as seen by the solid line in the figure. The response at low speeds, however, did not follow the trend line. The response at speeds between ± 6 pixels/second depended on the mean luminance of the screen as was also the case for the NDM pixels due to light-induced leakage.

Figure 6.17c shows the response of the NDS sensor while the spatial frequency and the speed of the sinusoidal stimulus were varied. The spatial frequency was varied from 0.08 to 0.225 cycles/pixel and the speed was varied from -30 to 30 pixels/second. The image representation of the data is similar to the NDS contour plot presented in Chapter 5. For a range of spatial frequencies (0.12 to 0.225 cycles/pixel), the response of the NDS pixels remains fairly constant at a given speed (as seen by the white contour lines in Figure 6.17c). As explained for the NDM case, here too at very low speeds, the sensor was unable to detect speed of the sinusoidal grating and responded to the mean luminance of the screen. This is seen in the figure as concentric contour lines in the middle of the plot.

The average and the standard deviation of the NDS pixel response is plotted against the spatial frequency of the stimulus in Figure 6.17d. The lower trace represents the response of the NDS pixels at a speed of 8 pixels/second. It can be seen that the average response of the NDS pixels is fairly independent of the spatial frequency of the stimulus between 0.12 to 0.22 cycles/pixel. The upper trace is the response of the NDS pixels at a speed of 20 pixels/second. It can be seen that the response increases linearly with spatial frequency up to 0.16 cycles/pixel, and after that the increase with spatial frequency is small as indicated by the trend line in the figure.

The response of the NDS and NDM pixels presented above are consistent with the simulation results of the models presented in Chapter 5, although they are affected by light-induced leakage at low speeds. The results show that for a range of spatial frequencies, both the sensors respond linearly with speed.

6.4 Discussion

There were several limitations of this design which did not allow us to use this sensor on an autonomous system. We discuss them in this section for both the NDS and NDM pixels.

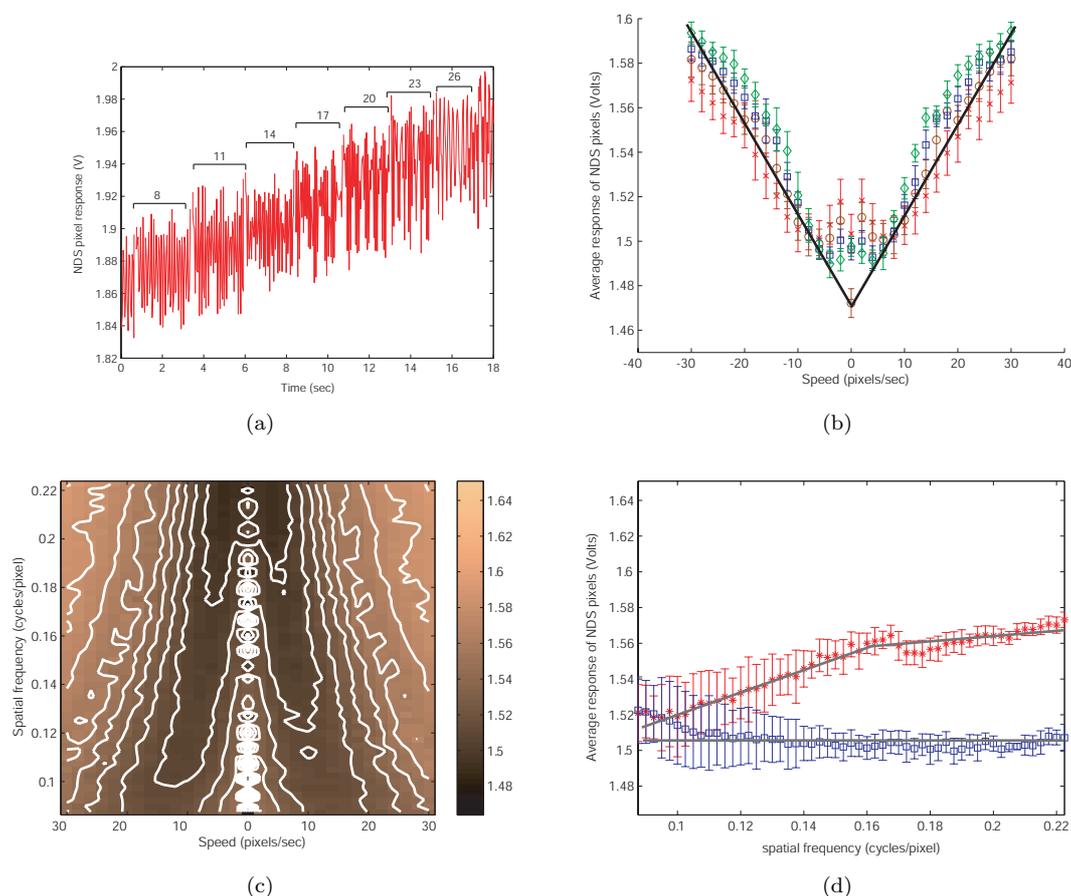


FIGURE 6.17. Response of NDS pixels to a sinusoidal stimulus. (a) A time trace of the response as stimulus speed is stepped from 8 to 26 pixels/second as indicated by brackets in the figure. (b) Response (mean and standard deviation) as the speed of the stimulus is varied. Multiple sets of data collected at spatial frequencies 0.14, 0.1675, 0.195, and 0.2225 cycles/pixel are overplotted to show that the response is relatively independent of spatial frequency. The solid line represents a speed-dependent linear output. (c) Contour plot of the average response as the spatial frequency and speed of the stimulus are varied. The magnitude increases from dark to light. The response magnitude increases with speed and is almost independent of the spatial frequency for a range of spatial frequency as shown by the vertical portion of the contour lines. (d) Average response and standard deviation against the spatial frequency. The upper trace shows the response at a speed of 20 pixels/second, while the lower trace shows the data at a speed of 8 pixels/second. The line segments show the trend of the data for each case.

The response of the sensor at very low speeds (between ± 6 pixels/second) depended on the mean luminance of the screen. This was found to be true for both the NDM and the NDS pixels. This was probably due to light-induced leakage in the pixels. The light incident on the chip can create a small photocurrent in the transistors of a pixel. The response of the chip was recorded for only a selected number of pixels; however, the leakage takes place across the chip. Therefore, even for small leakage currents in individual pixels the net effect is appreciable. Another reason that may have contributed to this sensitivity is the size of the transistors in the pixel. The chip was fabricated in a $0.18 \mu\text{m}$ process and therefore the size of transistors was small. This makes the transistors more susceptible to second order effects that cause the gate of a transistor to have less control over the channel current. This may have further deteriorated the response of the chip at small signal levels as is the case for small speeds.

The response of the NDS and NDM pixels were not much different from each other. From the analysis presented in Chapter 5, we had expected that the response of the NDS pixel would be more linear than the NDM pixel. This is also true for the chip and is seen via more equidistant contour lines in the case of the NDS pixels (Figure 6.17c). The distance between the contour lines in the case of the NDM pixels is smaller for speeds between ± 4 and ± 8 pixels/second, larger for speeds from ± 8 to ± 14 pixels/second, and then small again for higher speeds (see top half of Figure 6.16c). There was no appreciable difference in the spatial frequency tuning of the two pixels.

The fabricated chip also had a part of its scanner circuitry disabled due to a layout error. This did not allow us to choose a single pixel individually, but only a group of pixels in a row. This limited our testing capability of the sensor chip and we could not test the exact effect of biases on the output current of the pixels.

The above-mentioned problems may be solved by fabricating an improved version of this sensor. To tackle the problem of light induced leakage, the transistor sizes can be made larger so that their sensitivity to the second order effects is diminished. This would also reduce the mismatch of transistors across the chip. Also, the non-photoactive regions in the chip can be better shielded against light-induced leakage by using multiple layers of metal. The layout error in the scanner circuit may be easily corrected to give us greater control over the individual pixels on the chip. We can also fabricate an on-chip bias-generation circuit (Delbrück and Van Schaik, 2005) that could provide stable bias voltages to all the subcircuits in the sensor. This would greatly improve the usability of the sensor on a mobile platform. With these improvements, the sensor would be able to respond linearly to speed at lower speeds and spatial frequencies.

In the next section, we present an outline of how the non-directional speed output of this sensor may be used for navigation by a robotic platform.

6.5 Non-directional speed based navigation

The non-directional speed signal obtained from the NDM or NDS pixels may be used to control the path of a terrestrial robot. We consider the case of navigation in obstacle-ridden terrain by an autonomous mobile platform. A diagram of such a mobile system is shown in Figure 6.18. Three sensor chips are mounted with lenses for focusing the image of the scene onto the sensor die. The chips are oriented to compute image speed on the left, center, and right side of the moving platform. An onboard power supply PS is used to provide power to the chip and other electronics in the system. An onboard electronic circuit board CB has a microcontroller that generates a control signal based on the inputs from the sensor chips. This signal is sent to the servo motors that control the speed and orientation of the mobile platform.

The image speed on the sensor chip is computed by summing the response from all the pixels on a chip. Due to motion parallax, the local non-directional speeds estimated by the pixels would be larger for closer objects and smaller for farther objects. A sum of all the pixels on the right side is compared with the sum of all the pixels on the left side. For two carefully calibrated sensor chips,

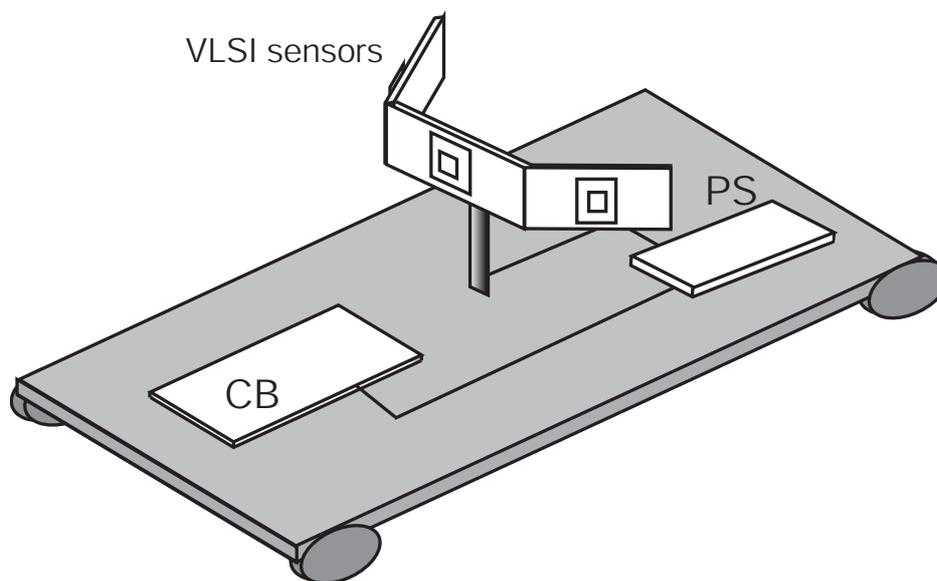


FIGURE 6.18. Diagram showing the arrangement of sensor chips on a mobile platform. Three sensor chips are each equipped with a lens to focus an image onto the chips and are shown in the diagram oriented such that they compute image speed on the left, center, and right side of the mobile platform. The motors moving the mobile platform are controlled by a circuit board CB. The power supply to all the electronics is provided by an onboard supply PS.

this difference will reveal which sensor computed a higher image speed value. The microcontroller then uses the difference between the output of the two sensors to turn the platform away from the side which witnessed a higher speed (Serres *et al.*, 2006). For the case of static obstacles that are not directly ahead, this would provide an elegant yet simple mechanism for autonomous navigation.

For an obstacle that is directly ahead of the robot, the central sensor chip is utilized. This chip computes the net image speed of the central portion of the scene and sends this information to the microcontroller. The microcontroller compares the response of the central chip with the response of the left and right sensors. If the response of the central chip is smaller than either of the other two sensors, it does not affect the navigation strategy of the robot. However, any time the speed response of the central chip exceeds the response of both the left and right sensor chips, the microcontroller is programmed to regard it as a frontal obstacle. Upon discovering this, the mobile platform immediately slows down and makes a turn at a predefined angle, preferably 90° , to avoid collision. This strategy will ensure that the robot does not collide head-on with an obstacle. For obstacles on the left and right side, the robot would navigate smoothly around them.

Such a robot may be very useful in terrain where population density is very low and most of the obstacles are static, for example for a search mission on a desolate island, or more interestingly on an exploration mission on Mars.

6.6 Summary

We have designed and fabricated an analog VLSI chip based on the NDM and the NDS models. To achieve this, we designed and tested a novel add-multiply circuit, characterization data from which was presented. The speed-dependent output from the chip was recorded and analyzed. The results from the fabricated chip follow the response characteristics of the two non-directional models.

This suggests that the non-directional chip may be used to compute image speed. The surrounding scanner circuitry allows for a summation of the pixel responses. This spatial averaging cancels out individual fluctuations in a single pixel's output due to process-induced variations. Thus, similar to an insect visual system, this sensor produces a robust estimate of the image speed independent of local variations. This neuromorphic architecture can be employed as a sensor on an autonomous robot as was discussed in the previous section. We described how the signals from three chips mounted on the robot may be used to navigate it in an obstacle-ridden environment.

In the final chapter of this dissertation, we present a brief discussion about the performance challenges of the tracking-based and non-directional models described in this dissertation, and present directions for future work.

CHAPTER 7

FINAL COMMENTS AND FUTURE WORK

In this dissertation, we have presented neurobiologically-inspired models that may be used in the design of autonomously navigating platforms. We have presented a collision avoidance system that utilizes tracking to improve the performance of existing biological models. Earlier work on this topic was based on mathematical analysis of optical flow based algorithms to generate tracking parameters for a monocular agent (Fermuller and Aloimonos, 1992, 1993). We have experimentally shown by implementing a camera-based system under which conditions the performance of a biological collision avoidance model is most affected, and how we may utilize tracking to improve the performance, specifically for the Rind and the STI models. We have also implemented a novel analog VLSI sensor that can measure the image speed in a scene independent of a range of spatial frequencies. This speed information may further be used for controlling the orientation and speed of an autonomous robot.

In this chapter, we present our final remarks on the design issues related to the discussed biomimetic models and propose improvements to the work presented in this dissertation.

7.1 Summary of contributions

We have mathematically analyzed two representative collision avoidance models, the Rind model and the STI model, and have pinpointed reasons behind their failure in different collision scenarios. We have proposed solutions for the limitations in the models and have implemented a collision avoidance system based on these algorithms. Our solution relies upon extracting the 3D motion information about the obstacle by tracking its trajectory in the two-dimensional visual field. This approach is consistent with those utilized in the computer vision community. However, the novelty of our approach is that the collision avoidance algorithms we implemented are derived from spatio-temporal frequency based biological models. These models are robust to noisy visual inputs (Adelson and Bergen, 1985) as compared to optical flow based algorithms used in the computer vision community which can be mathematically unstable. Hence, spatio-temporal frequency based schemes are more reliable under real world conditions. By combining the biological models with tracking, we have shown that our implementation of the Rind and STI models with tracking perform consistently better and are a major improvement over the Rind and STI models without tracking.

We have also simplified two non-directional speed estimation units derived from the neuronally based EMD model such that they respond to a wider range of speed and spatial frequency of the visual stimulus than the original models. We have implemented these modified NDS and NDM models as an analog VLSI chip. The characterization results from the two pixel types show that their outputs are relatively independent of the spatial frequency of the visual stimulus while being linearly dependent on the stimulus speed. This speed dependent response may be used for simple navigational schemes as have been described to model the landing and centering response of the honeybees (Srinivasan *et al.*, 1993, 2000).

In the next few sections, we discuss some of the shortcomings of the models described in this dissertation. We also propose improvements to these models that may address these shortcomings as a possible future work of this dissertation.

7.1.1 Collision avoidance models with tracking

The collision avoidance models with tracking capability that we have presented in this dissertation show marked improvement over the Rind and STI models without tracking. However, we tested

this system only for controlled collision scenarios. For those cases, our simple tracking algorithm was able to reliably track the approaching object. Specifically, we have adopted a simple tracking algorithm that computes the position of the obstacle based on the color of the object present in the visual field. This scheme may be generalized to implement the same algorithm to compute the position based on the brightness pattern in the visual field. However, in all these cases the inherent assumption is that either the color or the brightness of the obstacle stands out from its environment. This may not always be true. In such cases, a more involved tracking algorithm may be required. We have experimented with one such algorithm based on a small-field tracking system in the fly brain (Reichardt *et al.*, 1989; Higgins and Pant, 2004b). This algorithm tracks a single object in the visual field based on the relative motion of the object and the background. This algorithm was not implemented in our system due to the extra computational time required to process it. When implemented, this made our system operate at a very low frame rate and therefore, respond only to small speeds. A possible future improvement of this model would be to investigate how to effectively implement a better tracking algorithm without severely slowing our system.

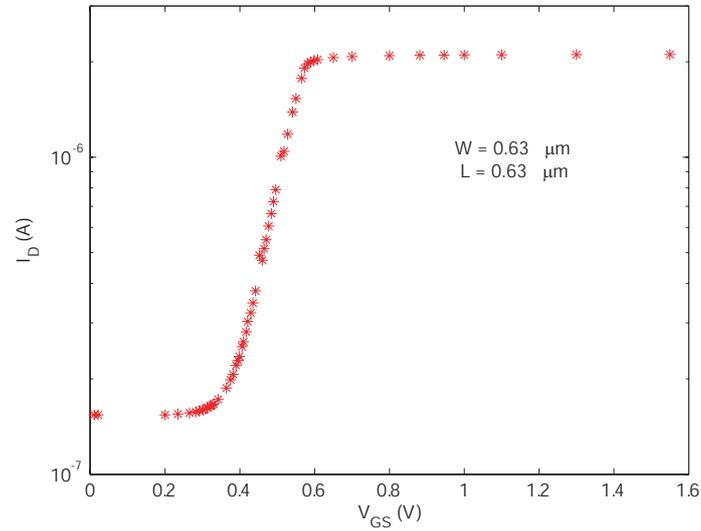
We present some ideas to improve the existing tracking algorithm to make the collision detection system cope with more complicated scenarios. One solution would be to use a sensor chip in parallel with our system that implements the small-field tracking algorithm. We have earlier designed an analog VLSI chip that can track small moving targets (Higgins and Pant, 2004a). This chip may be used in parallel with the collision avoidance algorithm to divide the computational load of the system. We can also implement the Rind and STI models as analog sensor chips and thereby reduce the power consumption due to carrying a laptop on our mobile platform.

Another desirable improvement on this system would be to implement a tracking algorithm that can predict the position of the target based on its past positions. In a control loop, the loop delay severely hampers the performance of such a system. Our implementation is no exception. Using a sophisticated Kalman filter based predictive algorithm (Kalman, 1960), this loop delay may be reduced significantly. We must note, however, that this algorithm would require additional computational machinery. The best result would be produced if this algorithm is incorporated within a VLSI chip. An effort can be made to combine the small-field tracking model mentioned above with a predictive algorithm. Such an algorithm would use the small-field tracking model to compute the target location and feed it to a Kalman filter that would generate a control signal. This would greatly improve the existing system and make it more usable for complex real world scenarios.

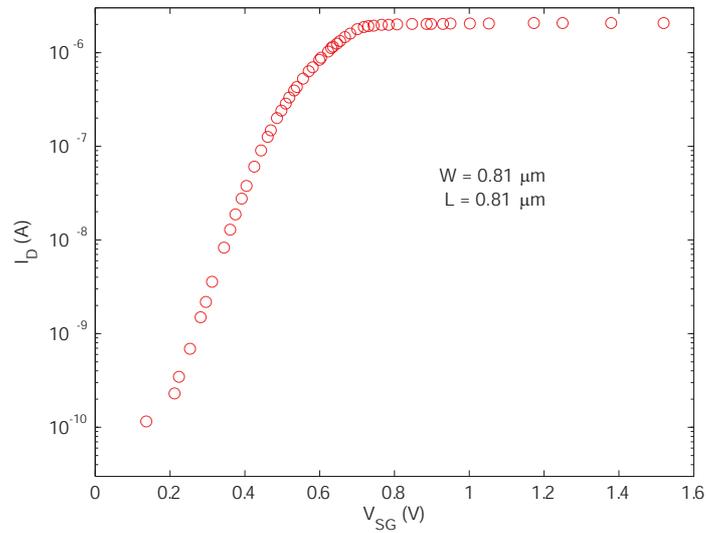
7.1.2 Non-directional speed-estimation model

The non-directional models described in this dissertation present a simple scheme to reliably estimate the speed of a visual stimulus. However, their responses are contrast dependent. For the simplified NDS model, the output is linearly dependent on the contrast of the object. The output of the simplified NDM model is dependent on contrast squared. This strong dependence on the contrast of the visual stimulus will incorrectly indicate an increase in the speed when only the contrast of the stimulus increases. This limitation may be addressed by incorporating a contrast saturating nonlinearity at the level of the phototransduction circuit. This nonlinearity will saturate the input to the subsequent processing stages when the contrast of the visual stimulus increases or decreases from a predefined range. A contrast saturation model like a hyperbolic tangent function may be incorporated in the analog chip. This unit may be inserted between the adaptive photoreceptor circuit and the summation/multiplication circuits in the simplified NDS/NDM pixels. A circuit analog of the saturating non-linearity is a transconductance amplifier with unity gain. The differential mode range of this amplifier may be chosen such that it suppresses variations due to the contrast of the visual stimulus except for a very small range where the response is linear with contrast.

Another severe limitation was encountered in the VLSI implementation of the two non-directional models. The VLSI chip was fabricated in a 0.18 μm CMOS process. For this technology, the



(a)



(b)

FIGURE 7.1. Short-channel effect on MOS transistors. a) Gate-to-source voltage (V_{GS}) versus the drain current (I_D) plot for an NMOS transistor of size $0.63 \mu\text{m} \times 0.63 \mu\text{m}$. b) Source-to-gate voltage (V_{SG}) versus the drain current (I_D) plot for a PMOS transistor of size $0.81 \mu\text{m} \times 0.81 \mu\text{m}$.

minimum separation between the source and the drain of a MOS transistor on the chip, known as the *channel length*, is set at $0.18\ \mu\text{m}$. At such a small channel length, the second order effects of the device become very pronounced, especially the effect of minimum current flowing through the transistor or the *leakage current*. In our design, some of the transistors were only three to four times the minimum length. While measuring the characteristics of an NMOS and a PMOS transistor on our chip, we found that at such small channel lengths, the minimum current flowing through the transistor was determined by the leakage current, and not by the gate-to-source voltage which otherwise regulates this current. This is due to the *short-channel effects* (Taur and Ning, 1998).

Figure 7.1 illustrates this problem in a small NMOS test transistor and a slightly longer PMOS test transistor. These test transistors were fabricated on the same die as our sensor chip. The channel length of the NMOS device was only $0.63\ \mu\text{m}$. We find that the drain current (I_D) of the NMOS transistor is approximately constant at 1.5×10^{-7} A, for a change of three hundred millivolts of gate-to-source voltage, V_{GS} (see Figure 7.1a). This problem is almost negligible in the case of a slightly longer PMOS device with a channel length of $0.81\ \mu\text{m}$. As seen in Figure 7.1b, the drain current varies over four orders of magnitude with the source-to-gate voltage (V_{SG}). No reliable data could be collected for V_{SG} less than a hundred millivolts because of the limitation of our measuring apparatus which was not sensitive below 10^{-10} A. This indicates that at this length, the PMOS transistor was free of the short-channel effect.

Based on the two characterization curves for NMOS and PMOS transistors shown in Figure 7.1, we can be reasonably confident that we will not witness the short-channel effect if we design all the transistors in our chip to be longer than $0.81\ \mu\text{m}$. All the device ratios may be scaled up after this adjustment to the size of the shortest transistors. This would make the chip functional over a larger range of variations to the input illumination.

The VLSI sensor that we designed had two type of pixels each with a low-pass filter circuit that was not utilized. The chip also had an interleaved array of NDM and NDS pixels that could not be used simultaneously. As future work, this chip may be designed for only one type of pixel and with no low-pass filter circuit. The removal of the low-pass filter circuit will provide us with extra space in each pixel that may be utilized to make bigger photodiodes. The bigger a photodiode the less sensitive it is to the nonlinear effect of currents on its edges. Thus a bigger photodiode will provide a more reliable transduced signal to the processing circuitry in a pixel. After re-designing the chip, it may be utilized to compute relative depth map by motion parallax. This chip may then be tested in the real world by using it as a sensor in a closed-loop system for navigation.

7.2 Conclusions

Biomimetic models provide engineers with alternative approaches to otherwise complex and seemingly intractable problems. In this dissertation, we have engineered a VLSI sensor chip for visual speed estimation and designed a collision avoidance system that actively tracks the obstacles in its visual field. As future work to this thesis, the speed estimation sensor chip may be elaborated to implement the STI or the Rind model. This sensor chip may also be integrated with a tracking chip to simultaneously track and avoid obstacles. The results from our experiments with the collision avoidance system may also be used to design behavioral experiments for insects. These behavioral experiments may be used to study the role of tracking during an avoidance maneuver in insects. An autonomous system can be made ‘smarter’ by utilizing biological strategies like learning from the past successes and failures. Other biologically inspired strategies like combining information from several sensors (e.g. audio, visual, sonar) may also be incorporated in future designs. In conclusion, a deeper and more comprehensive understanding of the biological systems is needed to propel the design ideas for future intelligent systems.

REFERENCES

- Adelson, E. H. and J. R. Bergen (1983). Spatiotemporal energy models for the perception of motion. *J Opt Soc Amer* 73: 1861–1862.
- Adelson, E. H. and J. R. Bergen (1985). Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A-Opt. Image Sci. Vis.* 2: 284–299.
- Adiv, G. (1985). Determining 3-dimensional motion and structure from optical flow generated by several moving-objects. *IEEE Trans. Pattern Anal. Mach. Intell.* 7: 384–401.
- Aloimonos, Y., I. Weiss, and A. Bandopadhyay (1988). Active vision. *International Journal of Computer Vision* 2: 333–356.
- Ancona, N. and T. Poggio (1995). Optical flow from 1D correlation – Application to a simple time-to-crash detector. *Int. J. Comput. Vis.* 14: 131–146.
- Andreou, A.G., K.A. Boahen, P.O. Pouliquen, A. Pavasović, R.E. Jenkins, and K. Strohhahn (1991). Current mode subthreshold MOS circuits for analog VLSI neural systems. *IEEE Trans. Neural Networks* 2: 205–213.
- Bajcsy, R. (1985). Active perception vs. passive perception. In *Proc. IEEE Workshop on Computer Vision*, pp. 55–59.
- Bandopadhyay, A. and D.H. Ballard (1991). Egomotion perception using visual tracking. *Computational Intelligence* 7: 39–47.
- Blanchard, M., F. C. Rind, and P. F. M. J. Verschure (2000). Collision avoidance using a model of the locust LGMD neuron. *Robot. Auton. Syst.* 30: 17–38.
- Boahen, K. A. (1998). Communicating neuronal ensembles between neuromorphic chips. In *Neuromorphic systems engineering: neural networks in silicon*, pp. 229–259.
- Borst, A. (1990). How do flies land? *Bioscience* 40(2): 292–299.
- Borst, A. and S. Bahde (1986). What kind of movement detector is triggering the landing response of the housefly? *Biol. Cybern.* 55: 59–69.
- Borst, A. and M. Egelhaaf (1989). Principles of visual-motion detection. *Trends Neurosci.* 12: 297–306.
- Bruckstein, A., R.J. Holt, I. Katsman, and E. Rivlin (2005). Head movements for depth perception: Praying mantis versus pigeon. *Autonomous Robots* 18: 21–42.
- Bulthoff, H., J. Little, and T. Poggio (1989). A parallel algorithm for real-time computation of optical-flow. *Nature* 337: 549–553.
- Cuadri, J., G. Lina, R. Stafford, M.S. Keil, and E. Roca (2005a). A bioinspired collision detection algorithm for VLSI implementation. Vol. 5839, pp. 238–248.
- Cuadri, J., G. Lina, R. Stafford, M.S. Keil, and E. Roca (2005b). Collision videos. <http://www.staff.ncl.ac.uk/richard.stafford/models.html>.
- Dacke, M. and M.V. Srinivasan (2007). Honeybee navigation: distance estimation in the third dimension. *Journal of Experimental Biology* 210: 845–853.

- Delbruck, T. (1993). Silicon retina with correlation-based, velocity-tuned pixels. *IEEE Trans. Neural Netw.* 4: 529–541.
- Delbrück, T. and C.A. Mead (1993). Analog VLSI phototransduction. Technical report 30, Computation and Neural Systems Program, California Institute of Technology.
- Delbrück, T. and A. Van Schaik (2005). Bias current generators with wide dynamic range. *Analog Integrated Circuits and Signal Processing* 43(3): 247–268.
- DeWeerth, S.P. and T.G. Morris (1994). Analog VLSI circuits for primitive sensory attention. In *Proc. IEEE Int. Symp. Circuits and Systems*, Vol. 6, pp. 507–510.
- Duric, Z., A. Rosenfeld, and J. Duncan (1999). The applicability of Green’s theorem to computation of rate of approach. *International Journal of Computer Vision* 31(1): 83–98.
- Egelhaaf, M. (1985a). On the neuronal basis of figure-ground discrimination by relative motion in the visual system of the fly. I. Behavioral constraints imposed on the neuronal network and the role of the optomotor system. *Biol. Cybern.* 52: 123–140.
- Egelhaaf, M. (1985b). On the neuronal basis of figure-ground discrimination by relative motion in the visual system of the fly. II. Figure-detection cells, a new class of visual interneurons. *Biol. Cybern.* 52: 195–209.
- Egelhaaf, M. (1985c). On the neuronal basis of figure-ground discrimination by relative motion in the visual system of the fly. III. Possible input circuitries and behavioral significance of the FD-cells. *Biol. Cybern.* 52: 267–280.
- Egelhaaf, M. (1990). Spatial interactions in the fly visual system leading to selectivity for small-field motion. *Naturwissenschaften* 77: 182–185.
- Egelhaaf, M. and A. Borst (1993). A look into the cockpit of the fly – visual orientation, algorithms, and identified neurons. *J. Neurosci.* 13: 4563–4574.
- Egelhaaf, M., A. Borst, A. K. Warzecha, S. Flecks, and A. Wildemann (1993). Neural circuit tuning fly visual neurons to motion of small objects. II. input organization of inhibitory circuit elements revealed by electrophysiological and optical-recording techniques. *J. Neurophysiol.* 69: 340–351.
- Etienne-Cummings, R., S. Fernando, N. Takahashi, V. Shtonov, J. Spiegel, and P. Mueller (1993). A new temporal domain optical flow measurement technique for focal plane VLSI implementation. In Bayoumi, M., L. Davis, and K. Valavanis, editors, *Proc. Comp. Arch. Machine Perception*, pp. 241–250.
- Etienne-Cummings, R., J. Spiegel, and P. Mueller (1996). A visual smooth pursuit tracking chip. *Advances in Neural Information Processing Systems* 8: 706–712.
- Etienne-Cummings, R., J. Spiegel, P. Mueller, and M. Z. Zhang (2000). A foveated silicon retina for two-dimensional tracking. *IEEE Trans. Circuits Syst. II-Analog Digit. Signal Process.* 47: 504–517.
- Exner, S. (1894). Entwurf zu einer physiologischen Erklärung der psychischen Erscheinungen. *I. Teil* pp. 37–140.
- Fermuller, C. and Y. Aloimonos (1992). Tracking facilitates 3-d motion estimation. *Biol. Cybern.* 67: 259–268.
- Fermuller, C. and Y. Aloimonos (1993). The role of fixation in visual motion analysis. *International Journal of Computer Vision* 11(2): 165–186.

- Foley, J.D., J. Hughes, A. van Dam, and S. Feiner (1995). *Computer Graphics: Principles and Practice*. Addison-Wesley Professional.
- Franz, M. O. and H. G. Krapp (2000). Wide-field, motion-sensitive neurons and matched filters for optic flow fields. *Biol. Cybern.* 83: 185–197.
- Franz, M.O. and H.G. Krapp (1998). Wide-field, motion-sensitive neurons and optimal matched filters for optic flow. *Biological Cybernetics* 83(3).
- Gabbiani, F., H.G. Krapp, and G. Laurent (1999). Computation of object approach by a wide-field, motion-sensitive neuron. *Journal of Neuroscience* 19: 1122–1141.
- Gabbiani, F., C. Mo, and G. Laurent (2001). Invariance of angular threshold computation in a wide-field looming-sensitive neuron. *Journal of Neuroscience* 21(1): 314–329.
- Gibson, J. J. (1950). *The Perception of the Visual World*. Houghton Mifflin Co., Boston.
- Gilbert, B. (1974). A high-performance monolithic multiplier using active feedback. *IEEE J. Solid-State Circuits* SC-9: 267–276.
- Harris, R., D. O’Carroll, and S. Laughlin (2000). Contrast gain reduction in fly motion adaptation. *Neuron* 28(2): 595–606.
- Harrison, R.R. (2005). A biologically inspired analog IC for visual collision detection. *IEEE Transactions on Circuits and Systems-I* 52: 2308–2318.
- Harrison, R.R. and C. Koch (1998). An analog VLSI model of the fly elementary motion detector. In Jordan, M.I., M.J. Kearns, and S.A. Solla, editors, *Advances in Neural Information Processing Systems 10*, pp. 880–886. MIT Press.
- Hassenstein, B. and W. Reichardt (1956). Systemtheoretische analyse der Zeit-, Reihenfolgen- und Vorzeichenbewertung bei der Bewegungspertzeption des Rüsselkäfers *Chlorophanus*. *Zeitschrift für Naturforschung* 11b: 513–524.
- Hatsopoulos, N, F Gabbiani, and G Laurent (1995). Elementary computation of object approach by a wide-field visual neuron 270: 1000–1003.
- Higgins, C. M. and C. Koch (1999). Multi-chip neuromorphic motion processing. In *Proc. of the 20th Conference on Advanced Research in VLSI*, Atlanta, GA.
- Higgins, C.M. (2004). Nondirectional motion may underlie insect behavioral dependence on image speed. *Biological Cybernetics* 91: 326–332.
- Higgins, C.M., J.K. Douglas, and N.J. Strausfeld (2004). The computational basis of an identified neuronal circuit for elementary motion detection in dipterous insects. *Visual Neuroscience* 21: 567–586.
- Higgins, C.M. and V. Pant (2004a). A biomimetic VLSI sensor for visual tracking of small moving targets. *IEEE Transactions on Circuits and Systems I* 51(12): 2384–2394.
- Higgins, C.M. and V. Pant (2004b). An elaborated model of fly small target tracking. *Biological Cybernetics* 91(6): 417–428.
- Higgins, C.M., V. Pant, and R. Deutschmann (2005). Analog VLSI implementation of spatio-temporal frequency tuned visual motion algorithms. *IEEE Transactions on Circuits and Systems I* 52(3): 489–502.

- Higgins, C.M. and S.A. Shams (2002). A biologically-inspired modular VLSI system for visual measurement of self-motion. *IEEE Sensors Journal special issue on Integrated Multi-Sensor Systems and Signal Processing* 2(6): 508–528.
- Horiuchi, T., W. Bair, B. Bishofberger, J. Lazzaro, and C. Koch (1992). Computing motion using analog VLSI chips: an experimental comparison among different approaches. *International Journal of Computer Vision* 8: 203–216.
- Horiuchi, T., J. Lazzaro, A. Moore, and C. Koch (1991). A delay-line based motion detection chip. In Lippman, R.P., J.E. Moody, and D.S. Touretzky, editors, *Advances in Neural Information Processing Systems*, pp. 406–412. Morgan Kaufmann, San Mateo, CA.
- Horiuchi, T. and E. Niebur (1999). Conjunction search using a 1-D, analog VLSI-based attentional search/tracking chip. In *Proc. of the Conference on Advanced Research in VLSI*.
- Horn, B. (1986). *Robot Vision*, chapter 12. MIT Press, Cambridge, MA.
- Horn, B.K.P. and B.G. Schunck (1981). Determining optical flow. *Artificial Intelligence* 17: 185–203.
- Huber, S.A. and H. H. Bülthoff (1998). Simulation and robot implementation of visual orientation behaviors of flies. *From animals to animats, proceedings of the fifth conference on the simulation of adaptive behaviour* 5: 77–85.
- Huber, S.A. and H.H. Bülthoff (2002). Visuomotor control in flies and behavior-based agents. In R. J. Duro, J. Santos and M. Granas, editors, *Biological Inspired Robot Behavior Engineering, Studies in Fuzziness and Soft Computing*, pp. 89–117. Physica-Verlag, Heidelberg.
- Indiveri, G., J. Kramer, and C. Koch (1996). System implementations of analog VLSI velocity sensors. *IEEE Micro* 16: 40–49.
- Indiveri, G., P. Oswald, and J. Kramer (2002). An adaptive visual tracking sensor with a hysteretic Winner-Take-All network. *IEEE* pp. 324–327.
- Indiveri, G., A. M. Whatley, and J. Kramer (1999). A reconfigurable neuromorphic VLSI multi-chip system applied to visual motion computation. In *Proceedings of the 7th International Conference on Microelectronics for Neural, Fuzzy and Bio-inspired Systems*.
- Kalayjian, Z. and A. G. Andreou (1997). Asynchronous communication of 2D motion information using Winner-Take-All arbitration. *Analog Integr. Circuits Process.* 13: 103–109.
- Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering* 82: 35–45.
- Koch, C., J. Marroquin, and A. Yuille (1986). Analog “neuronal” networks in early vision. *Proc. Natl. Acad. Sci., USA* 83: 4263–4267.
- Kramer, J., R. Sarpeshkar, and C. Koch (1995). An analog VLSI velocity sensor. In *Proc. Int. Symp. Circuit and Systems (ISCAS)*, pp. 413–416, Seattle, WA.
- Krapp, H.G., B. Hengstenberg, and R. Hengstenberg (1998). Dendritic structure and receptive-field organization of optic flow processing interneurons in the fly. *J. Neurophysiol.* 79: 1902–1917.
- Krapp, HG and R Hengstenberg (1996). Estimation of self-motion by optic flow processing in single visual interneurons. *Nature* 384: 463–466.

- Kumar, N., W. Himmelbauer, G. Cauwenberghs, and A. G. Andreou (1998). An analog VLSI chip with asynchronous interface for auditory feature extraction. *IEEE Trans. on Circuit and Systems II* 45(5): 600–606.
- Land, M. F. and R. D. Fernald (1992). The evolution of eyes. *Annu. Rev. Neurosci.* 15: 1–29.
- Land, M.F. and T.S. Collett (1974). Chasing behaviour of houseflies (*Fannia canicularis*): description and analysis. *J. Comp. Physiol.* 89: 331–357.
- Landolt, O., A. Mitros, and C. Koch (2001). Visual sensor with resolution enhancement by mechanical vibrations. *Proceedings 2001 Conference on Advanced Research in VLSI* pp. 249–264.
- Laurent, G. and F. Gabbiani (1998). Collision-avoidance: Nature’s many solutions. *Nature Neuroscience* 1(4): 261–263.
- Lazzaro, J. and J. Wawrzynek (1997). Speech recognition experiments with silicon auditory models. *Analog Integr. Circuits Process.* 13: 37–51.
- Lazzaro, J., J. Wawrzynek, M. Mahowald, M. Sivilotti, and D. Gillespie (1993). Silicon auditory processors as computer peripherals. *IEEE Trans. Neural Networks* 4(3).
- Lee, D., N. (1980). The optic flow field: the foundation of vision. *Philos. Trans. R. Soc. London Series B* 290: 169–179.
- Lindemann, J.P., R. Kern, J.H. Hateren, H. Ritter, and M. Egelhaaf (2005). On the computations analyzing natural optic flow: Quantitative model analysis of the blowfly motion vision pathway. *The Journal of Neuroscience* 25(27): 6435–6448.
- Liu, S.C. (1996). Silicon model of motion adaptation in the fly visual system. In *Proceedings of the Third Joint Caltech/UCSD Symposium on Neural Computation*. Pasadena, CA.
- Mahowald, M. and C. Mead (1991). The silicon retina. *Scientific American* 264(5): 76–82.
- Mallot, H. A., H. H. Bulthoff, J. J. Little, and S. Bohrer (1991). Inverse perspective mapping simplifies optical-flow computation and obstacle detection. *Biol. Cybern.* 64: 177–185.
- Mead, C. A. and T. Delbrück (1991). Scanners for visualizing activity of analog VLSI circuitry. Technical report 11, California Institute of Technology, Department of Computation and Neural Systems.
- Mead, C.A. (1989). *Analog VLSI and Neural Systems*. Addison-Wesley, Reading, MA.
- Milde, J.J. and N.J. Strausfeld (1990). Cluster organization and response characteristics of the giant fiber pathway of the blowfly *Calliphora erythrocephala*. *J. Comp. Neurol.* 294: 59–75.
- Miller, K. T. and G. L. Barrows (1999). Feature tracking linear optic flow sensor chip. *IEEE Int’l Symp. Ckts. Sys.* pp. 116–119.
- Ozalevli, E. and C.M. Higgins (2005). Reconfigurable biologically-inspired visual motion systems using modular neuromorphic VLSI chips. *IEEE Transactions on Circuits and Systems I* 52(1): 79–92.
- Pant, V. and C.M. Higgins (2007). A biomimetic focal plane speed computation architecture. In *Computational Optical Sensing and Imaging (COSI) conference*, Vancouver, Canada.
- Poggio, T. and A. Verri (1987). Against quantitative optical flow. In *In Proc. 1st ICCV*, London.

- Poggio, T., W. Yang, and V. Torre (1989). *Optical flow: computational properties and networks, biological and analog*, pp. 355–370. Addison-wesley Publishing Co. Inc.
- Prazdny, K. (1980). Egomotion and relative depth map from optical flow. *Biological Cybernetics* 36: 87–102.
- Reichardt, W. (1961). Autocorrelation, a principle for the evaluation of sensory information by the central nervous system. In Rosenblith, WA, editor, *Sensory Communication*, pp. 303–317. MIT Press, New York.
- Reichardt, W., M. Egelhaaf, and A.K. Guo (1989). Processing of figure and background motion in the visual-system of the fly. *Biol. Cybern.* 61: 327–345.
- Reichardt, W., M. Egelhaaf, and R.W. Schögl (1988). Movement detectors provide sufficient information for local computation of 2-D velocity field. *Naturwiss.* 75: 313–315.
- Reichardt, W., T. Poggio, and K. Hausen (1983). Figure-ground discrimination by relative movement in the visual system of the fly. II. Towards the neural circuitry. *Biol. Cybern.* 46: 1–30.
- Reynolds, C. (1999). Steering behaviors for autonomous characters. In *Proc. Game Developers Conf.*, pp. 763–782, San Francisco, CA.
- Rind, F.C. (1984). A chemical synapse between two motion detecting neurones in the locust brain. *Journal of Experimental Biology* 110(1): 143–167.
- Rind, F.C. and D.I. Bramwell (1996). Neural network based on the input organization of an identified neuron signaling impending collision. *J. Neurophysiology* 75: 967–985.
- Rind, F.C. and P.J. Simmons (1992). Orthopteran DCMD neuron: A reevaluation of response to moving objects. I. Selective responses to approaching objects. *J. Neurophysiology* 68: 1654–1666.
- Rivera-Alvidrez, Z. (2005). Computational modeling of neurons involved in fly motion detection. Master’s thesis, University of Arizona (Advisor, C.M. Higgins).
- Rivera-Alvidrez, Z. and C.M. Higgins (2005). Contrast saturation in a neuronally-based model of elementary motion detection. *Neurocomputing* 65-66: 173–179.
- Santer, R.D., R. Stafford, and F.C. Rind (2004). Retinally-generated saccadic suppression of a locust looming-detector neuron: Investigations using a robot locust. *Journal of Royal Soc. London Interface* .
- Sarpeshkar, R., W. Bair, and C. Koch (1993). An analog VLSI chip for local velocity estimation based on Reichardt’s motion algorithm. In Hanson, S.J., J. Cowan, and L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pp. 781–788, San Mateo, CA. Morgan Kaufmann.
- Sarpeshkar, R., J. Kramer, G. Indiveri, and C. Koch (1996). Analog VLSI architectures for motion processing: From fundamental limits to system applications. *Proc. IEEE* 84: 969–987.
- Serres, J., F. Ruffier, and N. Franceschini (2006). Two optic flow regulators for speed control and obstacle avoidance. In *Proceedings of the first IEEE Int. Conference on Biomedical Robotics and Biomechatronics*, pp. 750–757, Pisa, Italy.
- Simmons, P.J. and F.C. Rind (1992). Orthopteran DCMD neuron: A reevaluation of response to moving objects. II. Critical cues for detecting approaching objects. *J. Neurophysiology* 68: 1667–1682.

- Single, S. and A. Borst (1998). Dendritic integration and its role in computing image velocity. *Science* 281: 1848–1850.
- Srinivasan, M.V., M. Lehrer, W.H. Kirchner, and S.W. Zhang (1991). Range perception through apparent image speed in freely-flying honeybees. *Visual Neuroscience* 6: 519–535.
- Srinivasan, M.V., S. Zhang, M. Lehrer, and T. Collett (1996). Honeybee navigation en route to the goal: visual flight control and odometry. *Journal of Experimental Biology* 199(1): 237–244.
- Srinivasan, M.V. and S.W. Zhang (1997). Visual control of honeybee flight. In Lehrer, M, editor, *Orientation and communication in arthropods*, pp. 95–113.
- Srinivasan, M.V., S.W. Zhang, J.S. Chahl, E. Barth, and S. Venkatesh (2000). How honeybees make grazing landings on flat surfaces. *Biological Cybernetics* 83: 171–183.
- Srinivasan, M.V., S.W. Zhang, and K. Chandrashekara (1993). Evidence for two distinct movement-detecting mechanisms in insect vision. *Naturwissenschaften* 80: 38–41.
- Stafford, R., R.D. Santer, and F.C. Rind (2007a). A bio-inspired collision mechanism for cars: Combining insect inspired neurons to create a robust system. *Biosystems* 87(2-3): 164–171.
- Stafford, R., R.D. Santer, and F.C. Rind (2007b). A bio-inspired visual collision detection mechanism for cars: Combining insect inspired neurons to create a robust system. *Biosystems* 87: 164–171.
- Stocker, A. (2004). Analog VLSI focal-plane array with dynamic connections for the estimation of piecewise-smooth optical flow. *IEEE Trans. on Circuit and Systems* 51(5): 963–973.
- Strausfeld, N.J. (1976). *Atlas of an Insect Brain*. Springer-Verlag, Berlin.
- Sun, H. and B.J. Frost (1998). Computation of different optic variables of looming objects in pigeon nucleus rotundus neurons. *Nat Neurosci* 1: 296–303.
- Tammero, L.F. and M.H. Dickinson (2002a). Collision-avoidance and landing responses are mediated by separate pathways in the fruit fly, *Drosophila melanogaster*. *The Journal of Experimental Biology* 205: 2785–2798.
- Tammero, L.F. and M.H. Dickinson (2002b). The influence of visual landscape on the free flight behavior of the fruit fly *Drosophila melanogaster*. *The Journal of Experimental Biology* 205: 327–343.
- Taur, Y. and T.H. Ning (1998). *Fundamentals of Modern VLSI Devices*. Cambridge University Press.
- Toumazou, C., F.J. Lidgey, and D.G. Haigh (1990). *Analogue IC design: the current-mode approach*. Peter Peregrinus Ltd.
- van Santen, J.P.H. and G. Sperling (1985). Elaborated Reichardt detectors. *J. Opt. Soc. Am. A-Opt. Image Sci. Vis.* 2: 300–321.
- Verri, A., F. Girosi, and V. Torre (1990). Differential techniques for optical-flow. *J. Opt. Soc. Am. A-Opt. Image Sci. Vis.* 7: 912–922.
- Verri, A. and T. Poggio (1989). Motion field and optical-flow - qualitative properties. *IEEE Trans. Pattern Anal. Mach. Intell.* 11: 490–498.
- Verschure, Paul F.M.J. (1998). Distributed adaptive control: Explorations in robotics and the biology of learning. *Informatik/Informatique* 1: 25–29.

Virsik, R. and W. Reichardt (1976). Detection and tracking of moving objects in the fly *Musca domestica*. *Biological Cybernetics* 23: 83–98.

Warren, W.H. and D.J. Hannon (1988). Direction of self-motion is perceived from optical-flow. *Nature* 336: 162–163.

Weber, K., S. Venkatesh, and M.V. Srinivasan (1997). Insect inspired behaviours for autonomous control of mobile robots. In Srinivasan, M.V. and S. Venkatesh, editors, *From living eyes to seeing machines*, Oxford. Oxford University Press.

Zhang, K. C., M. I. Sereno, and M. E. Sereno (1993). Emergence of position-independent detectors of sense of rotation and dilation with hebbian learning - an analysis. *Neural Comput.* 5: 597–612.

Zigmond, J. Michael, E.F. Bloom, C.L. Story, J.L. Roberts, and L.R. Squire (1999). *Fundamental Neuroscience*. Academic Press, San Diego, CA.