

DISS. ETH NO. 20736

Bio-Inspired Optical Flow Vision Sensors for Visual Guidance of Autonomous Robots

A dissertation submitted to the
ETH ZURICH

for the degree of
Doctor of Sciences

presented by
RICO MÖCKEL
Dipl. Ing., University of Rostock
born 02. Februar 1981
citizen of Germany

accepted on the recommendation of
Prof. Dr. Rodney Douglas, examiner
Dr. Shih-Chii Liu, co-examiner
Prof. Dr. Tobias Delbrück, co-examiner
Prof. Dr. Auke Jan Ijspeert, co-examiner

2012

*to my wonderful wife
Roswitha
for her love and patience
and
to my little sunshine
Pia
for making
my life worth living*

Acknowledgement

There are many people I owe thanks since without them this thesis would never have been possible:

First of all I would like to thank Dr. Shih-Chii Liu. I am grateful both for her support and advice throughout my entire Ph.D. project and for the great freedom she gave me to develop and pursue my own ideas.

I am indebted to Professor Rodney Douglas, Professor Tobias Delbrück and Professor Auke Jan Ijspeert for participating in my thesis committee. It were Dr. Shih-Chii Liu and Professor Tobias Delbrück who shared and encouraged my passion for circuit design. It was Professor Auke Jan Ijspeert who supported me during a most desperate time.

This thesis was carried out at the Institute of Neuroinformatics (INI) which belongs both to the University of Zurich and ETH Zurich. I am grateful to these institutions and to both the directors Professor Rodney Douglas, Professor Kevan Martin, and Prof. Richard Hahnloser for allowing me being part of this great multi-disciplinary institute with such an inspiring atmosphere. I would like to express my thanks as well to my colleagues and friends at INI especially Yingxue Wang and Prashanth D'Souza.

I would further like to thank Stéphane Bonardi, Massimo Vespignani, and Alexander Spröwitz from the EPFL Biorobotics Laboratory for their kind support.

My thesis is part of a joint project on "Bioinspired Vision-based Microflyers" together with the Laboratory of Intelligent Systems (LIS) at the École Polytechnique Fédérale de Lausanne (EPFL). I am very grateful to the director of the LIS, Prof. Dario Floreano and his first assistant Dr. Jean-Christophe Zufferey who initiated this project and established our fruitful partnership. Dr. Zufferey provided me with lots of valuable insights into the control of unmanned micro aerial vehicles and advice. I would further like to thank my Ph.D. colleagues at the EPFL Antoine Beyeler and Adam Klaptocz for all the exciting joined work.

I would probably never have worked at the INI without Prof. Ralf Salomon from the

University of Rostock who established my first contact to the INI. I would also like to thank him for his guidance and valuable advice.

I wish to thank the Swiss National Science Foundation (grant number: 200021-105545/1) as well as the University of Zurich, the ETH Zurich, and the EPFL for supporting my thesis project with funding.

I would like to express my thanks to the scientists from whose work this thesis highly benefited from. I am grateful to Professor Nicolas Franceschini and Dr. Franck Ruffier for their work on autonomous flyers as well as their elementary motion detector model. I had some fruitful discussions with them during the Symposium on Flying Insects and Robots in 2007. I deeply thank the late Dr. Jörg Kramer on whose work my dynamic optic flow sensor is based on, Dr. Jean-Christophe Zufferey for his inspiring work on microflyers, Professor Mandyam Srinivasan for his image interpolation algorithm and his advice during discussions at different occasions, Dr. Piotr Dudek for his work on single instruction multiple data processors based on switched current circuits and Prof. Tobias Delbrück for his work on vision circuits like his dynamic vision sensor as well as his bias generator circuits. I am grateful to Prof. Reid R. Harrison for discussions on his motion detection chip during the IEEE Symposium on Circuits and Systems in 2007 and to Professor John G. Harris and Dr. Xin Qi for the additional information that they gave me on their time-to-first-spike imager. I would like to thank Professor Ralph Etienne-Cummings for discussions on his various motion detection chips and active pixel imagers, Dr. Victor Gruev for the clarifications on the correlated double sampling technique for current-mode imagers and Dr. Mark Tilden for his useful advice and the background information on the robots he developed for WowWee that he gave me during the 2009 Telluride Neuromorphic Cognition Engineering Workshop.

I would like to deeply thank Martin Lanz, Peter Brühlmeier and André Badertscher. These technicians helped not only with bonding of chips and PCB design and fabrication but also with lots of very valuable advice.

Furthermore, I would like to express my thanks to all the highly motivated students that I had the pleasure to supervise during their semester or master project at the Institute of Neuroinformatics - Andreas Bur, Judith Felder, Stephan Weiss, Roger Jäggi and David Jäckel.

The latex template developed for this thesis is with some self-made modifications based on two templates one by Melissa Penny and another created by Marc Wüest and Jörg Mäder from the Institute for Atmospheric and Climate Science at the ETH Zurich.

I am especially grateful to my parents as well as my grand parents who supported and

encouraged me not only during my studies and Ph.D. project but throughout my whole life, as well as to my brother Daniel.

Finally, I wish to thank my wife Roswitha for her understanding, support and love.

Abstract

Visual perception - the ability to interpret the environment by sensing light - is an impressive ability of biological organisms. Vision allows animals and human beings to explore and to interact with their environment. It is a daily task biological organisms seem to perform with very little effort. Even smallest insects with a body size in the range of millimeters depend on vision as a main source of feedback for their behavior control which includes impressive flight and landing skills. Despite the efforts of many researchers in mimicking these skills in artificial systems, biological vision systems are still more powerful, versatile and efficient than any artificial systems developed so far. Nature has evolved integrated and compact solutions where the vision system is well-adjusted to the properties of the sensory-motor control system and the physical body of the biological organism - properties that artificial systems still lack.

Optical flow is a major visual sensor cue used by animals in navigating their environment. Optical flow is due to the relative motion of the vision system and the objects in its environment. It encodes information about the shape and position of objects as well as the self-motion of the animal. This information is used extensively in particular by insects such as flies and bees in flight control.

Because of the impressive properties of biological systems, various groups have designed and implemented sensors in analog Very Large Scale Integrated (aVLSI) technology that mimic biological optical flow processing on various levels of abstraction. In parallel with recent efforts to construct miniature aerial platforms, optic flow information from camera sensors together with gyroscope outputs are used on such platforms to demonstrate properties such as autonomous flight.

At the EPFL Laboratory of Intelligent Systems, Jean-Christophe Zufferey and others developed a 10-gram indoor airplane and demonstrated autonomous wall avoidance. The airplane uses an off-the-shelf one-dimensional image sensor from which optical flow caused by the relative motion of objects on the left and right side of the airplane is extracted. Optical flow is computed on a microcontroller that is also running the control

loops steering the airplane's behavior.

Analog VLSI optical flow sensors can allow such an airplane to leave the controlled conditions of a lab environment and to free the microcontroller from the optical flow computation. These sensors implement circuits both for phototransduction - the conversion of light into electrical signals - and optical flow processing. At the Institute of Neuroinformatics several researchers developed monolithic optical flow sensors. Jörg Kramer for example presented a suite of time-of-travel sensors that extract contrast edges from the visual stimuli presented to the sensor and successfully extract optical flow by measuring their time-of-travel - the time these contrast edges require to travel the distance between adjacent sensor pixels.

This thesis contributes to the evaluation and design of optical flow sensors suitable for miniature flying platforms, and to investigations into optical flow algorithms and flight control strategies suitable for control of autonomous robots.

This thesis addresses a major drawback of existing time-of-travel sensors: the sensors presented so far showed optical flow outputs that depended both on stimulus velocity and stimulus contrast making the interpretation of these outputs for steering autonomous robots challenging. This contrast dependence is caused because the properties of the event signals that are generated when contrast edges are detected depend on stimulus contrast. As a first achievement this thesis presents a novel edge detection circuit that does not suffer from this problem. The circuit uses two thresholds to trade the minimum stimulus contrast that is required to emit an edge detection event and the maximum amplitude of signal noise that is rejected. The novel edge detection circuit is demonstrated as part of a one-dimensional time-of-travel optical flow sensor implemented in aVLSI technology. The algorithm implemented on this sensor design was chosen after careful consideration of the existing optical flow sensors and the intended flying platform. The sensor adapts over 4 orders of background light intensities and due to the novel edge detection circuit extracts optical flow information for stimulus contrasts down to 2.5%. Measurements from the sensor are presented when being operated on a rotating platform. Furthermore, the sensor is demonstrated in a closed-loop control application for steering a simulated car.

The second achievement is the creation of a miniaturized optical flow sensor module. The combination of analog VLSI vision sensors with classical machine vision techniques is challenging but can be beneficial for autonomously flying micro aerial vehicles (MAV) and autonomous robots in general. This thesis presents a continuous-time custom aVLSI vision sensor that implements circuits for phototransduction, filtering and amplification of photoreceptor signals. The sensor is adapting over four orders of back-

ground light intensities. In combination with a plastic lens a 0.3-gram vision sensor module for miniature aerial platforms is created which is sufficiently lightweight so that 3 vision sensor modules can be carried by small MAVs like the 10-gram airplane by Zuffery et. al. Optical flow is computed on a separate microcontroller using the Image Interpolation Algorithm by M. V. Srinivasan that integrates the optical flow across sensor pixel. Experiments are presented where similarly as suggested by Zufferey et al. the optical flow extracted from the output signals of this vision sensor is compared to the readings from a gyroscope. Comparisons between the outputs of both sensors show encouraging results suggesting that the adaptive vision sensor is well suited for visual guidance of autonomous robots.

As a third achievement this thesis presents an analysis of the current state-of-the-art control strategy where the distance-dependent translational optical flow component is extracted by measuring the total flow with an optical flow sensor and subtracting the rotational component estimated by a gyroscope. The analysis concludes that although the currently used airplanes and control strategy are successful, distance estimation with optic flow sensors is more efficient when the autonomous robots mimic the flight behavior demonstrated by flies. These insects show flight behavior that alternates between periods of straight flight - best suited for extraction of the distance to obstacles with optical flow - and fast turns - commonly referred to as flight saccades. The reproduction of this flight behavior by artificial systems is challenging. MAVs that are capable of performing saccades need to be used. Such MAVs are typically passively unstable and thus would need to be actively stabilized but the quality of the distance estimates extracted from optic flow sensors as well as the maneuverability of the autonomous systems would largely benefit.

With the presentation of novel optical flow sensor solutions, the analysis of existing sensors, MAVs and control strategies this thesis provides important steps towards more powerful optical flow guided autonomous robots.

Zusammenfassung

Biologische Organismen haben die beeindruckende Fähigkeit ihre Umgebung allein auf Basis von visuellen Reizen wahrnehmen zu können. Sehen erlaubt Tieren und Menschen ihre Umgebung zu erforschen und mit ihr zu interagieren. Es ist eine Tätigkeit, die biologischen Organismen scheinbar ohne nennenswerten Aufwands jeden Tag verrichten. Sogar kleine Insekten mit einer Körpergrösse von wenigen Millimetern verwenden visuelle Reize zur eigenen Verhaltenssteuerung. Dieses Verhalten schliesst die beeindruckenden Flug- und Ladefähigkeiten der Insekten ein. Trotz des grossen Aufwandes, den eine Vielzahl von Wissenschaftlern betreiben, um diese Fähigkeiten mittels künstlicher Systeme zu kopieren, sind biologische visuelle Wahrnehmungssysteme immer noch viel leistungsstärker, vielseitiger und effizienter als jedes bisher entwickelte künstliche System. Mittels Evolution hat die Natur integrierte und kompakte Lösungen geschaffen, in denen die visuelle Wahrnehmungssysteme an die Sensor-Motor-Kontrollsysteme und die physikalischen Körpereigenschaften perfekt angepasst sind. Diese Anpassung fehlt bei künstlichen Systemen noch immer.

Eine wichtige visuelle Informationquelle, die Tiere verwenden, um sich in einer Umgebung zu orientieren ist Optischer Fluss. Dieser entsteht durch die relative Bewegung des Wahrnehmungssystems zu den Objekten in der Umgebung. Optischer Fluss enthält Informationen sowohl über die Form und Position der Objekte als auch über die Eigenbewegung des Tieres. Diese Informationen werden insbesondere von Insekten wie Fliegen und Bienen exzessiv für die Flugsteuerung verwertet.

Die beeindruckenden Fähigkeiten von biologischen Systemen haben eine Vielzahl von Forschungsgruppen dazu veranlasst, Sensoren in Form von hochintegrierten Systemen zu kreieren und implementieren, welche die Form der Verarbeitung des Optischen Flusses von biologischen Systemen auf unterschiedlichen Abstraktionsebenen kopieren. Parallel zu den kürzlich erfolgten Investitionen in die Konstruktion von Miniaturflugkörpern, wurde von Kameras generierter Optischer Fluss mit Messungen von einem Gyroskop kombiniert um autonomen Flug zu demonstrieren.

Am Labor für Intelligente System des EPFL haben J. C. Zufferey und andere ein 10-Gramm Innenraumflugzeug entwickelt, das beim Fliegen selbstständig Zusammenstöße mit den Wänden des Raumes vermeidet. Das Flugzeug verwendet dazu eine eindimensionale Standardkamera aus deren Bildern der von Objekten zur rechten und linken Seite des Flugzeuges verursachter Optischer Fluss berechnet wird. Die Berechnung übernimmt dabei derselbe Mikroprozessor, der auch den Kontrollalgorithmus des Flugzeuges beherbergt.

Analoge hochintegrierte Sensoren zur Messung des Optischen Flusses könnten es einem solchen Flugzeug erlauben die kontrollierte Umgebung eines Labors zu verlassen und dazu beitragen den Mikroprozessor von der aufwendigen Aufgabe der Berechnung des Optischen Fluss zu befreien. Diese Sensoren integrieren Schaltkreise, die Licht in elektrische Signale umwandeln und aus diesen den Optischen Fluss berechnen. Am Institut für Neuroinformatik haben mehrere Wissenschaftler monolithische Sensoren zur Messung des Optischen Flusses entwickelt. J. Kramer zum Beispiel hat eine Vielzahl von sogenannter "time-of-travel" Sensoren präsentiert, die erfolgreich den Optischen Fluss bestimmen, indem sie die Zeit messen, die eine Kontraständerung benötigt um zwei benachbarte Sensorpixel zu passieren.

Diese Doktorarbeit leistet Beiträge zur Analyse und zum Design von Sensoren zur Messung des Optischen Flusses bei Miniaturflugkörpern und zur Untersuchung von Algorithmen zur Messung des Optischen Flusses und zur Steuerung von autonomen Robotern.

Diese Doktorarbeit befasst sich mit einem wichtigen Nachteil derzeit existierender "time-of-travel" Sensoren: Die bisher vorgestellten Sensoren verfügen über einen Messausgang, dessen Wert sowohl von der Geschwindigkeit der visuellen Stimuli als auch von deren Kontrast abhängt. Diese Abhängigkeit macht die Interpretation der Messwerte und die Verwendung der Sensoren zum Ansteuern von autonomen Robotern schwierig. Die Kontrastabhängigkeit beruht darauf, dass die Eigenschaften der elektrischen Signale, die bei Erkennung einer Kontraständerung erzeugt werden, von der Amplitude der Kontraständerung anhängen. Eine erste Errungenschaft dieser Doktorarbeit ist eine neue Schaltung zur Erkennung von Kontraständerungen, die nicht unter diesem Problem leidet. Die Schaltung verwendet zwei Schwellwerte mit denen sich die minimale Kontraständerung, die benötigt wird, um ein Änderung am Schaltungsausgang zu erzeugen, sowie die maximale Rauschamplitude, die ausgesondert wird, einstellen lassen. Die Schaltung zur Detektion von Kontraständerungen wird als Teil eines eindimensionalen hochintegrierten "time-of-travel" Sensors zur Messung des Optischen Flusses demonstriert. Der für diesen Sensor verwendete Algorithmus wurde

nach sorgfältigem Studium bereits existierender Sensoren zur Optischen Fluss Messung und der fliegenden Plattform, für die der Sensor geplant wurde, ausgewählt. Der Sensor passt sich an die sich verändernde Hintergrundbeleuchtung über vier Dekaden an. Dank des neuen Schaltkreises zur Detektion von Kontraständerungen misst der Sensor Optischen Fluss für Kontraständerungen bis runter zu 2.5%. Sensordaten werden präsentiert, die gemessen wurden, während der Sensor auf einer rotierenden Plattform getestet wurde. Ausserdem wird der Sensor als Teil einer Regelungsschleife zur Steuerung eines computersimulierten Autos demonstriert.

Die zweite Errungenschaft ist die Erzeugung eines miniaturisierten Sensormodules zur Messung des Optischen Flusses. Die Kombination von analogen hochintegrierten visuellen Sensoren mit klassischen rechnergestützten Verfahren zur Bildverarbeitung ist kompliziert aber auch lohnenswert für autonome miniaturisierte Flugmaschinen und autonome Roboter im Allgemeinen. Diese Doktorarbeit präsentiert einen massgefertigten analogen hochintegrierten Sensor zur visuellen Wahrnehmung, der Schaltkreise für die Umwandlung von Licht in elektrische Signale sowie deren Filterung und Verstärkung implementiert. Der Sensor passt sich an die sich verändernde Hintergrundbeleuchtung über vier Dekaden an. In Kombination mit einer Plastiklinse entsteht ein 0.3 Gramm Sensor zur visuellen Wahrnehmung für miniaturisierte Flugkörpern. Das geringe Gewicht des Sensormodules erlaubt eine einfache Integration von mindestens 3 Sensormodulen auf miniaturisierte Flugmaschinen wie dem 10 Gramm Flugzeug von Zufferey und anderen. Der Optische Fluss wird auf einem separaten Mikroprozessor unter Verwendung des Bildinterpolationsalgorithmus von M. V. Srinivasan berechnet. Dieser Algorithmus integriert den Optischen Fluss über mehrere Sensorpixel. Es werden Experimente präsentiert, in denen - ähnlich wie von J. C. Zufferey und anderen vorgeschlagen - der vom Sensor extrahierte Optische Fluss mit den von einem Gyroskop gewonnenen Daten verglichen wird. Der Vergleich liefert ermutigende Resultate, die darauf hindeuten, dass der Sensor für die visuelle Steuerung von autonomen Robotern gut geeignet ist.

Als dritte Errungenschaft präsentiert diese Doktorarbeit eine Analyse der derzeit aktuellen Steuerungsstrategie, bei der die distanzabhängige durch Translation erzeugte Komponente des Optischen Flusses ermittelt wird, indem mittels eines Optischen Fluss Sensors der totale Fluss gemessen und anschliessend die durch ein Gyroskop ermittelte Komponente des Optischen Flusses, die durch Rotation entsteht, subtrahiert wird. Die Analyse kommt zu dem Schluss, dass obwohl die derzeit verwendeten Flugzeuge und Steuerungsstrategie erfolgreich sind, die Distanzbestimmung mit Sensoren zur Messung des Optischen Flusses effizienter wäre, wenn die autonomen Roboter das Flugverhalten von Fliegen kopieren würden. Diese Insekten zeigen ein Flugverhalten,

bei dem sich Zeiten, in denen die Fliege geradeaus bewegt - was optimal zur Distanzbestimmung mit Optischem Fluss ist -, mit Zeiten abwechseln, in denen die Fliege schnelle Drehungen - sogenannte Saccaden - ausführt. Die Reproduktion dieses Flugverhaltens durch künstliche Systeme ist eine Herausforderung. Es müssten Miniaturflugkörper verwendet werden, die zur Ausführung von Saccaden fähig sind. Diese Flugkörper sind normalerweise passiv instabil und müssen aktiv stabilisiert werden. Dafür wären die Distanzbestimmungen mit Sensoren zur Messung des Optischen Flusses von besserer Qualität und die autonomen Systeme wären wendiger.

Mit der Demonstration von neuartigen Sensorlösungen zur Messung des Optischen Flusses, der Analyse existierender Sensoren, Miniaturflugkörper und Steuerungsstrategien präsentiert diese Doktorarbeit wichtige Schritte in Richtung leistungsstärkerer autonomer Roboter, die mit Hilfe von Messungen des Optischen Fluss gesteuert werden.

Contents

Acknowledgement	v
Abstract	ix
Zusammenfassung	xiii
1 Introduction	1
1.1 The problem at hand	1
1.2 Related work	2
1.2.1 Optical flow in insects	2
1.2.2 Optical flow based control of MAVs	3
1.2.3 Optical flow and vision sensors for micro aerial vehicles	3
1.3 Motivation for VLSI optical flow sensors	5
1.4 Original contributions	6
1.5 Thesis organization	8
2 Which information can be extracted with optical flow?	11
2.1 Definition of optical flow and optical flow sensors	11
2.2 Formal description of optical flow perceived by a perfect spherical sensor	13
2.2.1 Optical flow induced by sensor's translation	13
2.2.2 Optical flow induced by sensor's rotation	17
2.3 Conclusions: theoretical properties and limitations of optical flow	18

3	Optical flow based control in flies and honeybees	25
3.1	Flies and bees regulate flight speed and wall distance through optical flow	25
3.2	Honeybees use optical flow to measure flight distances	26
3.3	Insects' compound eyes are optimized for perception of optical flow	31
3.4	Visual interneurons in blowflies estimate self-motion	31
3.5	Conclusions	34
4	Optical flow based guidance of micro aerial vehicles	39
4.1	Wall-avoidance with 10-gram indoor airplane and single linear camera . .	39
4.2	Obstacle avoidance with outdoor airplane and 7 optical mouse sensors .	43
4.3	Passing corridors with quadcopter and single camera with fish-eye lens .	46
4.4	Conclusion	48
5	A short introduction to analog VLSI technology	53
5.1	The basic elements for chip design: transistors, capacitors, resistors . . .	54
5.1.1	CMOS field effect transistors	54
5.1.2	Capacitors	61
5.1.3	Resistors	62
5.2	Mismatch in analog VLSI circuits	63
5.2.1	Impact, classification and causes of device mismatch in aVLSI . .	63
5.2.2	Mismatch compensation and avoidance in aVLSI	65
5.3	Noise	70
5.4	Conclusion	71
6	Evaluation of optical flow sensors for autonomous robots	73
6.1	Sensor optics, pixel size, and fill-factor	73
6.2	Evaluation criteria for optical flow algorithms and sensors	79
6.3	Evaluation of algorithms for optical flow sensors and their VLSI counterpart	82
6.3.1	Gradient-based intensity algorithm	86
6.3.2	Intensity-based correlation	89

6.3.3	Token-based correlation	90
6.3.4	Time-of-travel	91
6.4	Conclusions	95
7	Dynamic optical flow sensor	101
7.1	Dynamic optical flow sensor - overview, motivations and properties	102
7.1.1	Sensor overview	102
7.1.2	Properties of the facilitate-and-sample circuit implementation . . .	106
7.1.3	Sensor history, modifications and acknowledgment	110
7.1.4	Remarks on circuit layout and fabrication process	111
7.2	Circuits of the dynamic optical flow sensor	112
7.2.1	Adaptive logarithmic wide-dynamic-range photoreceptor circuit . .	113
7.2.2	LMC circuit	118
7.2.3	Double threshold edge detection circuit	124
7.2.4	Circuits for pulse shaping and generation of facilitate signal	130
7.2.5	Sample & hold & decay circuit	133
7.2.6	Scanner circuit	135
7.2.7	Direction selection circuit	139
7.3	Experimental results	139
7.3.1	Measured outputs at a pixel	140
7.3.2	Optic flow output from a single pixel	142
7.3.3	Optic flow output from all pixels	143
7.3.4	Contrast independence of optical flow output	148
7.3.5	Perception of self-rotation	149
7.4	Discussion	151
8	Steering a simulated car with the dynamic optical flow sensor	161
8.1	Experimental Setup	161
8.2	Single-layer perceptron controller	163
8.3	Experimental results	164

8.4	Discussion	166
9	Adaptive aVLSI vision sensor	169
9.1	Overview of the adaptive vision system	170
9.1.1	Sensor overview	170
9.1.2	Sensor history	172
9.2	Circuits of the dynamic optical flow sensor	173
9.2.1	Adaptive logarithmic wide-dynamic-range photoreceptor circuit . .	173
9.2.2	LMC circuit	175
9.2.3	Scanner circuit	176
9.3	Linear Image Interpolation Algorithm	177
9.4	Experiments	179
9.4.1	Experimental setup	179
9.4.2	Experimental results	181
9.5	Discussion	184
10	MAVs and controller for efficient use of optical flow sensors	199
10.1	Theoretical analysis on wall avoidance with optical flow guided MAVs . . .	200
10.2	Discussion	205
11	Conclusion	209
11.1	Main achievements	209
11.2	Future works	212
	List of Figures	219
	List of Tables	221
	Index	223
	Bibliography	224

Chapter 1

Introduction

1.1 The problem at hand

In 1925 Helmholtz described the exciting experience of an observer moving in the woods:

”Suppose, for instance, that a person is standing in a thick woods, where it is impossible for him to distinguish, except vaguely and roughly, in the mass of foliage and branches all around him what belongs to one tree and what to another, or how far apart the separate trees are, etc.

But the moment he begins to move forward, everything disentangles itself, and immediately he gets an apperception of the material contents of the woods and their relations to each other on space, just as if he were looking at a good stereoscopic view of it.

... objects that are at rest by wayside ... appear to glide past us in our field of view in the opposite direction to that in which we are advancing. More distant objects do the same way, only more slowly, while very remote bodies like the stars maintain their permanent positions in the field of view ... Evidently, under these circumstances, the apparent angular velocities of objects in the field of view will be inversely proportional to their real distances away; and consequently safe conclusions can be drawn as to the real distance of the body ...” (Helmholtz, 1925)

This description by Helmholtz showing how image motion allows determining the distance of objects and resolving their shapes in a complex scene was quoted by Gibson (1950) when he studied and described the richness of information encoded in image motion or **optical flow**. While both the observer and the objects in the observer’s environment are not moving it is fairly difficult for the observer to obtain a good perception

of the objects surrounding him despite the observer's ability for stereoscopic vision. But once the observer moves relative to the objects, the observer's vision system perceives image motion or optical flow and the observer experiences a plastic 3-dimensional view of the scene.

This thesis aims at the development of optical flow sensor solutions that allow autonomous robots exploiting optical flow to gain information about self-motion and the objects in their environment. The developed optic flow sensors aim especially at improving the visual guidance of micro aerial vehicles (MAVs). Such MAVs like the 10-gram airplane by our colleagues from the EPFL Laboratory of Intelligent Systems (Zufferey et al., 2006, 2007) have been recently developed to demonstrate autonomous flight control.

1.2 Related work

The phenomena of optical flow, its role in visual guidance of insects as well as its application in guiding micro aerial vehicles have been widely studied in the literature. However, the robustness and performance of biological vision systems have never been met by their engineered counterparts. This section gives a short overview of previously presented studies on insects, and on optical flow controlled MAVs and aVLSI optic flow sensors. More detailed descriptions can be found in chapters 3, 4 and 6. A detailed introduction to the concept of optical flow can be found in chapter 2.

1.2.1 Optical flow in insects

For a long time it is known that optical flow plays a major role in the visual guidance and flight control of insects like honeybees and flies. For example, honeybees can measure distances by means of the optical flow perceived during flight and communicate these distances to other bees as part of their waggle dance (Esch and Burns, 1996; Srinivasan et al., 2000a; Esch et al., 2001). Furthermore, honeybees have been shown to continuously reduce flight velocity during landing by keeping the perceived optical flow constant (Srinivasan et al., 2000b). When the ground gets closer to the insect, it appears to move faster. So by regulating optical flow, bees have an efficient mechanism to control flight speed during landing. It is further known that bees navigate through small tunnels by balancing the optical flow perceived by the left and right eye. By this mechanism bees fly in the middle of the tunnel keeping the distance to the walls equal

(Srinivasan and Zhang, 2004). Krapp and Hengstenberg (1996) have studied the visual interneurons of flies providing evidence that these cells are most sensitive to optical flow pattern generated during self-motion of the flies. Tammero and Dickinson (2002*b*) analyzed fruit flies flying in a cylindrical area showing that flies performed wall avoidance by turning away from the side of the eye receiving greater optical flow when the absolute optic flow value was above a certain threshold. Furthermore, David (1982) and Fry et al. (2009) showed that the flight speed of fruit flies (*Drosophila melanogaster*) is directly controlled by the perceived optical flow and is independent of wind speed. In summary, tiny insects like flies and bees have developed efficient and impressive flight strategies based on optical flow. These strategies allow them to fly autonomously despite their rather limited computational resources and rather primitive nervous systems.

1.2.2 Optical flow based control of MAVs

The success of optical flow based flight control in insects has inspired control engineers to model the insects' sensory-motor control (Rohrseitz and Fry, 2011) and to mimic flight strategies of insects with MAVs. For example, Green et al. (2003, 2004) have demonstrated automatic landing and obstacle avoidance with a 30-gram indoor flying robot based on insect-inspired control. Ruffier and Franceschini (2004, 2005, 2008) and Netter and Franceschini (2002) conducted studies on insect-inspired altitude control, take-off, and landing with tethered helicopters while Chahl et al. (2004) applied their discoveries about landing behavior in honeybees from (Srinivasan et al., 2000*b*) to outdoor unmanned aerial vehicles.

Our colleagues from the Laboratory of Intelligent Systems at EPFL have demonstrated autonomous flight control based on optic flow indoors and outdoors. For indoor flight, they used optical flow detected with the help of two linear standard cameras for wall avoidance on a 30-gram aircraft prototype (Zufferey and Floreano, 2005, 2006) as well as on a 10-gram airplane (Zufferey et al., 2006, 2007). They presented a novel optic flow based control algorithm called OptiPilot that they demonstrated on a 400-gram unmanned aircraft outdoors (Beyeler et al., 2009; Zufferey et al., 2010).

1.2.3 Optical flow and vision sensors for micro aerial vehicles

Although autonomous optical flow based flight control has been demonstrated, the performance of the presented solutions is still far behind those of insects which are also small, light-weight, consume little power and provide robust flight control under diffi-

cult light and wind conditions as well as in various terrains. There are many reasons behind this performance gap which include available materials, fabrication techniques, and control strategies of MAVs. Another important reason is the lack of optical flow sensors optimized for MAV flight control.

Designing sensors for optical flow estimation is already a challenge on its own. Vision sensors that are used on small MAVs furthermore have to be compact and light weight. Since computational power is limited, these sensors should ideally already compute optical flow or provide outputs that are well suited for estimating optical flow. Ideally sensors provide signals that can directly be fed into a controller. When being operated on MAVs optical flow vision sensors can face scenes where there is only limited texture, with low contrast, and where the background light varies over several decades.

Various sensors implemented in Very Large Scale Integrated (VLSI) technology which compute optical flow are described in (Moeckel and Liu, 2009) and chapter 6. The developed sensors can be roughly divided into three main categories:

1. Optical flow sensors like those presented by Mehta and Etienne-Cummings (2003, 2004, 2006) and Gruev and Etienne-Cummings (2004) combine **active pixel sensors (APS)** known from standard cameras with additional circuits that support the optical flow computation off-chip or that directly provide optical flow outputs. The use of APS allows the inclusion of standard camera on-chip compensation techniques like correlated double sampling providing reduced pixel fabrication mismatch (see section 5.2 for more details). Furthermore, these sensors typically have the advantage that they can be used as standard imagers. In fact, optical flow active pixel sensors inherit many of the properties of standard cameras: They provide a high spatial resolution but they typically do not feature the high dynamic range and the adaptation to background light at pixel level like many bio-inspired sensors of the third category.
2. The second group consists of **optical mouse sensors** designed to be operated in optical mouse devices for instance for controlling personal computers. These sensors can have various architectures including those of active pixel sensors. All optical mouse sensors have in common that they are typically optimized for measuring global optical flow under low-texture conditions making them interesting for optical flow based MAV control. The fact that standard optical mouse sensors can nowadays be purchased off-the-shelf and that they provide direct optical flow outputs makes them attractive for control of medium sized MAVs. Our colleagues from the Laboratory of EPFL Intelligent Systems and others have been using optical mouse sensors (Barber et al., 2005; Griffiths et al., 2007; Rodriguez et al.,

2007; Beyeler et al., 2009; Zufferey et al., 2010). However, when using optical mouse sensors one has to keep in mind that these devices are typically designed to be operated together with a LED that allows for constant and well-defined background illumination. This is why these sensors are typically lacking the ability to operate under challenging light conditions with high dynamic range as experienced in flight control indoors and outdoors in natural environments. Furthermore, when being purchased off-the-shelf there is only little customization possible allowing no access to local optical flow values since optical mouse sensors typically should provide only a single 2-D flow vector.

3. The third category comprises **bio-inspired aVLSI sensors** - optical flow and vision sensors that take inspiration from the visual system of insects and other animals with different levels of abstraction. The sensors presented by Harrison and Koch (1998, 1999), Miller and Borrows (1999), Higgins et al. (2005), and Liu (2000) that are made to **model the visual system of animals** like insects belong to this category. While these sensors are optimized to mimic particular biological aspects, they are usually not optimized for simple integration into MAVs. The sensors presented in this thesis highly take inspiration from previously described bio-inspired aVLSI sensor designs but are further optimized to allow a more simple integration into MAVs.

1.3 Motivation for VLSI optical flow sensors

I believe that robotic platforms like micro aerial vehicles can benefit from custom optical flow sensor implementations in Very Large Scale Integrated (VLSI) technology. This technology allows for the integration of circuits for both phototransduction - the conversion of light into electrical signals - as well as for the computation of optical flow on the same chip with dimensions in the range of millimeters. A single VLSI sensor can contain arrays of several hundred or thousands of pixels each processing optical flow in parallel and in continuous time. So by using VLSI optical flow sensors on MAVs, limitations of the on-board computational power in autonomous robots can be reduced. In addition, it is not necessary to transmit images from a camera on the MAV to an off-board computer to perform the optical flow computation and generation of the necessary control signals as demonstrated by Zingg et al. (2010) (section 4.3).

For instance the work by Beyeler et al. (2009) (presented in section 4.2) demonstrates how much MAVs can benefit from VLSI optical flow sensors. Beyeler et al. showed

that an airplane equipped with an array of seven optical mouse sensors, a differential pressure sensor for speed control and three rate gyroscopes for extracting the translational optical flow components can perform autonomous obstacle avoidance and altitude control. The ADNS-5050 off-the-shelf optical mouse sensors from Avago used in this demonstration are VLSI implementations that output two global optical flow outputs for the translation in x- and y-direction, respectively. The ADNS-5050 contains an image acquisition system and a digital signal processor for optical flow computation.

To the best of my knowledge, there has been no demonstration of fully autonomous flight control with the same autonomy as that shown by Beyeler et al. (2009) using bio-inspired optical flow sensors. The goal of this study is also to explain why bio-inspired optical flow sensors have not been so successful so far and how this can be changed. Autonomous MAVs could benefit from bio-inspired optical flow sensor implementations because of their properties such as local adaptation to background light, high bandwidth and low response time due to continuous-time operation of circuits performing computation in parallel, and low power consumption.

So far Harrison and Koch (1999) and Reichel et al. (2005) demonstrated bio-inspired optical flow sensors on wheeled robots. Green et al. (2004) mounted a bio-inspired optical flow detector on a 30-gram indoor airplane. Whenever the optical flow sensor detected an optical flow value exceeding a preset threshold they made the airplane turn away by fully deflecting the airplane's rudder. Furthermore, Green et al. (2003, 2004) demonstrated semi-autonomous landing on a basketball gym floor by keeping the perceived optical flow constant while reducing the airplanes forward velocity.

The sensor developed as part of this thesis also take inspiration from biology - mimicking certain biological principles but whenever there is the choice between modeling biology more precisely or designing more robust sensors optimized for controlling MAVs the design decision towards a more robust sensor solution is typically preferred. This is why the dynamic optic flow sensor (DOFS) presented in chapter 7 as well as the adaptive vision sensor (aVIS) presented in chapter 9 loosely belong to the category of bio-inspired aVLSI sensors but include additional circuits to allow a more easy integration into autonomous robots.

1.4 Original contributions

This thesis contributes to the field of VLSI circuits, optical flow vision sensors and their application in the control of autonomous robots. It advances the knowledge in the

following ways:

1. Chapter 6 presents the specification of criteria for optical flow sensors for steering flying platforms along with a comprehensive evaluation of the properties of existing optical flow sensors.
2. This thesis presents a novel edge detection circuit for the front-end of token based vision sensors (chapter 7). From the evaluation of existing flight controllers (chapter 4) and VLSI optical flow sensors (chapter 6) it can be concluded that optical flow output signals independent from stimulus contrast are preferred for autonomous flight control. However, as presented in chapter 6 many VLSI optical flow sensors show optical flow outputs that are dependent both on the velocity and contrast of visual stimuli. The presented edge detection circuit aims at overcoming this contrast dependence by generating reliable edge detection events.
3. This thesis presents a time-of-travel optical flow sensor for steering MAVs. The sensor contains the novel edge detection circuits for ensuring proper flow estimation even for low contrast stimuli down to 2.5% (chapter 7). The sensor's sensitivity to contrast is thus improved by a factor greater than 7 in comparison to previously published sensors with similar architectures like the ones presented by Kramer et al. (1995, 1997).
4. We developed an optical flow sensor module suited for ultra-lightweight flying robots. With a weight of only 0.3-gram it is one of the most lightweight of its kind. The module combines an adaptive aVLSI vision sensor with the Image Interpolation Algorithm (I2A) by Srinivasan (1993, 1994). The I2A extracts and spatially integrates the optical flow from the adaptive vision sensor. (chapter 9)
5. This thesis provides an analysis on the state-of-the-art optical flow control strategy where on constant-speed controlled MAVs the distance-dependent translational optical flow component is extracted by measuring the total flow with an optical flow sensor and subtracting the rotational component estimated by a gyroscope. Chapter 10 provides evidence that - although successful - the state-of-the-art control strategy does not optimally use the optical flow sensors. In contrast to MAVs, insects like honeybees and flies seem to have developed behavioral controllers that use optical flow sensors more efficiently by keeping the perceived optical flow constant instead of flight speed and by actively using flight behavior where periods of straight flight during which (almost) only translational optical flow is perceived

alternate with fast turns during which optical flow does not need to be measured (Schilstra and Hateren, 1999; Tammero and Dickinson, 2002a).

1.5 Thesis organization

This thesis is organized as follows.

Chapter 2 gives a detailed introduction to optical flow. It presents the definition of optical flow, derives its formal description, and studies the question of which information can be extracted by a perfect spherical optical flow sensor.

Chapter 3 analyzes how optical flow is used in nature for guiding the behavior of flying insects like flies and honeybees. It presents several strategies that flies and honeybees developed to extract information about self-motion and the environment using optical flow and to deal with possible imperfections in the extracted optical flow field.

Chapter 4 shows existing optical flow controlled flying platforms, their flight properties, controllers and which challenges these platforms present to optical flow sensors.

Chapter 5 introduces to the analog Very Large Scale Integrated (aVLSI) technology. It covers both the basic elements of aVLSI technology as well as challenges for the sensor design like device mismatch and circuit noise.

Chapter 6 specifies criteria for optical flow sensors for visual guidance of autonomous robots and gives a detailed analysis of existing optical flow sensors. This chapter motivates several design choices for the optical flow sensors presented in this thesis and reveals the challenges for the sensor design that arise from its application on micro aerial vehicles.

Chapter 7 presents the novel edge detection circuit as part of the dynamic optical flow sensor (DOFS). DOFS combines circuits that implement the facilitate-and-sample algorithm together with an adaptive logarithmic wide-dynamic-range photoreceptor front-end.

Chapter 8 demonstrates the capabilities and potentials of the dynamic optical flow sensor (DOFS) in a closed-loop application. DOFS is used for steering a simulated car in a simplified car driving simulation.

Chapter 9 presents the adaptive vision sensor (aVIS) implemented in VLSI technology. In combination with the Image Interpolation Algorithm (I2A) by Srinivasan (1993, 1994) a optical flow sensor module for steering micro aerial vehicles is formed.

Chapter 10 suggests that optical flow sensors can be used more efficiently if micro aerial vehicles (MAVs) and their controllers mimic the flight behavior found in insects more closely. It suggests that the performance gap between insects and their artificial counterparts should be closed from several directions - not only by designing specialized sensors but also by using MAVs that use the optical flow sensors as efficient as possible.

Chapter 11 concludes and describes future work.

Chapter 2

Which information can be extracted with optical flow?

Abstract

This chapter gives a detailed introduction to optical flow. It presents the definition of optical flow, derives its formal description, and studies the question of which information can be extracted by a perfect spherical optical flow sensor. Understanding the theoretical properties of a perfect optical flow sensor is important to know what are the limitations of any physical optical flow sensor and control strategies based on this type of sensor.

2.1 Definition of optical flow and optical flow sensors

Optical flow or optic flow is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene (Burton and Radford, 1978; Warren and Strelow, 1985; Wikipedia, 2011). In 1950 James J. Gibson studied and described the effects and properties of image motion caused by the relative motion of the observer and the scene and concluded that the image motion is a rich source of information about the observer's self-motion and the objects in the scene (Gibson, 1950). The fact that image motion caused by self-motion of an observer indicates the objects' distances from the observer has been known for a long time:

"... objects that are at rest by wayside ... appear to glide past us in our field of view in the

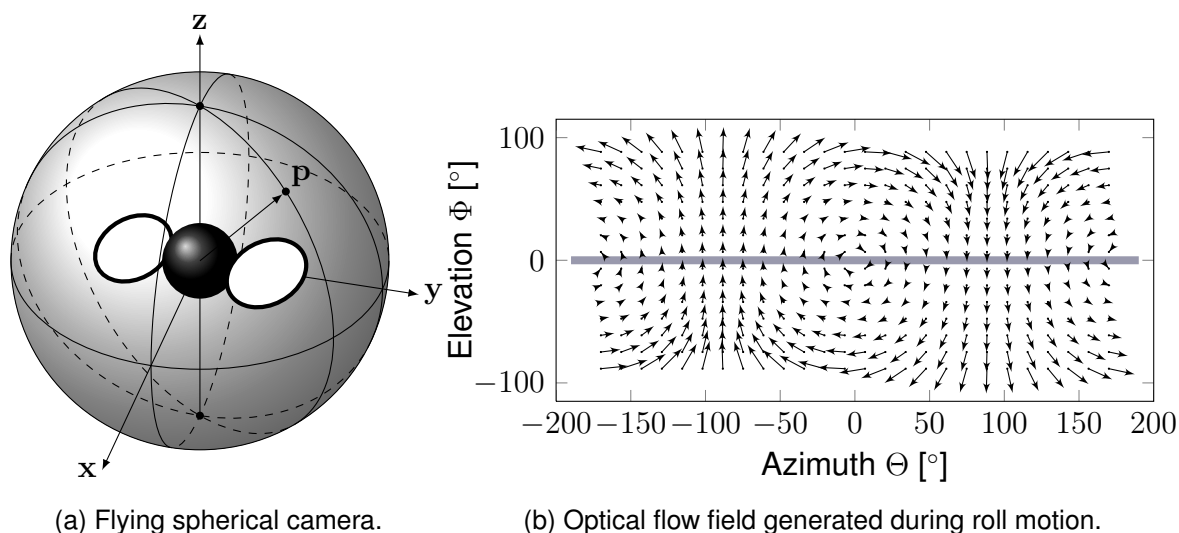


Figure 2.1: During roll motion (here rotation around the x-axis on left image) the flying spherical camera (black sphere on left image) sees the optical flow velocity field shown on the (right). p (left) is an arbitrary point in the environment being projected onto the camera.

opposite direction to that in which we are advancing. More distant objects do the same way, only more slowly, while very remote bodies like the stars maintain their permanent positions in the field of view ... Evidently, under these circumstances, the apparent angular velocities of objects in the field of view will be inversely proportional to their real distances away; and consequently safe conclusions can be drawn as to the real distance of the body ...” (Helmholtz, 1925).

Imagine a spherical camera like the one shown in figure 2.1a flying in an environment filled with objects. Each point on the camera image is the result of the projection of these objects in the 3-dimensional environment onto the 2-dimensional surface of the camera through an optical system. When the camera is moving relative to the objects, the image points on the camera image move as well. This image motion can be described by a velocity or **optical flow vector field** where each velocity vector points in the direction of motion of the corresponding image point with a vector length proportional to the magnitude of the motion velocity. Figure 2.1b depicts an example optical flow field here generated by roll motion of the spherical camera in figure 2.1a.

With **optical flow sensors** one seeks to extract the optical flow field from image motion. However, due to imperfection of the physical sensor - that in contrast to a perfect vision sensor requires a minimum amount of visual contrast, has limited spatial resolution, suffers from the aperture problem and noise in its photo-transduction (see chapter 6) -

2.2. Formal description of optical flow perceived by a perfect spherical sensor¹³

as well as due to limitations of the algorithms for optical flow extraction, the estimated optical flow typically does not perfectly correspond to the theoretically expected optical flow field.

To gain a better understanding of optical flow in the following section 2.2 all imperfections will be ignored and the formal description of optical flow perceived by a perfect spherical sensor during self-motion will be derived. This allows an analysis of the properties of perfect optical flow as well as of the theoretical limitations of optical flow based control (section 2.3). The impact of imperfections in the optical flow algorithm and sensor is discussed in chapter 6.

2.2 Formal description of optical flow perceived by a perfect spherical sensor

To gain a better understanding of what optical flow can provide to a control system, a formal description of the optical flow perceived during self-motion by a perfect spherical optical flow sensor is of benefit. A perfect sensor has infinite spatial resolution and does not suffer from any light intensity or contrast requirements. When the sensor is moving in a stationary environment, all optical flow is purely induced by the self-motion of the sensor. To simplify the mathematical equations in the following formal analysis it is assumed that the sensor is spherical with a radius $r = 1$ as indicated by the gray sphere in figure 2.2a.

Under these conditions every point \mathbf{p} in the environment of the sensor as well as the projection of that point onto the spherical surface of the sensor $\hat{\mathbf{p}}$ can be described in spherical coordinates by the two angles azimuth Θ and elevation Φ and the distance of the point from the center of the sensor $R(\mathbf{p}) = \|\mathbf{p}\|$ (figure 2.2a). Since $R(\hat{\mathbf{p}}) = \|\hat{\mathbf{p}}\| = r = 1$ the relation between the vector \mathbf{p} and its projection $\hat{\mathbf{p}}$ is given by

$$\hat{\mathbf{p}} = \frac{\mathbf{p}}{\|\mathbf{p}\|} \quad (2.1)$$

2.2.1 Optical flow induced by sensor's translation

In this section, equations describing the optical flow vector field caused by translation of the spherical sensor are derived. A Cartesian coordinate frame with its origin at

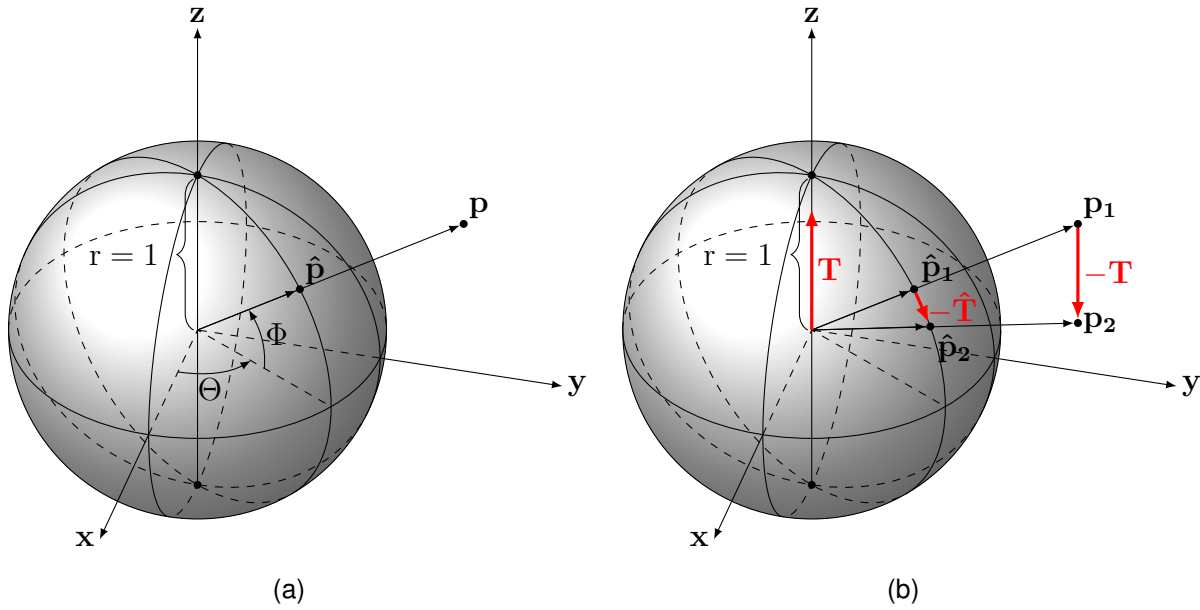


Figure 2.2: (a) \hat{p} is the projection of the point p onto a spherical vision sensor (plotted as a gray sphere) with radius $r = 1$. Each point on the sensor can be described by the two polar coordinates azimuth Θ and elevation Φ . (b) Displacement of the projection of point p_1 due to translation T of the unity sphere sensor. p_2 is the point's position after translation. \hat{p}_1 and \hat{p}_2 are the projections of p_1 and p_2 onto the sensor's surface, respectively.

the center of the spherical sensor as shown in figure 2.2b is assumed. If the sensor translates in space by an amount described by the translation vector T , any arbitrary point in the environment p_1 is shifted to a new position $p_2 = p_1 - T$. The projections of the original point \hat{p}_1 and after translation \hat{p}_2 are given by

$$\hat{p}_1 = \frac{\mathbf{p}_1}{\|\mathbf{p}_1\|} \quad (2.2)$$

$$\hat{p}_2 = \frac{\mathbf{p}_2}{\|\mathbf{p}_2\|} = \frac{\mathbf{p}_1 - \mathbf{T}}{\|\mathbf{p}_1 - \mathbf{T}\|} \quad (2.3)$$

$\hat{T} = \hat{p}_1 - \hat{p}_2$ is the projection of the point's translation onto the sensor's surface. However, for finding the trajectory that the projected point travels on the sensor's surface the precise shape of the sensor's surface has to be considered. Furthermore, since the optical flow vector field by definition is a velocity field a formal description of the velocity of the projected point perpendicular to the surface of the sensor \hat{v} (figure 2.3) has to be found.

2.2. Formal description of optical flow perceived by a perfect spherical sensor¹⁵

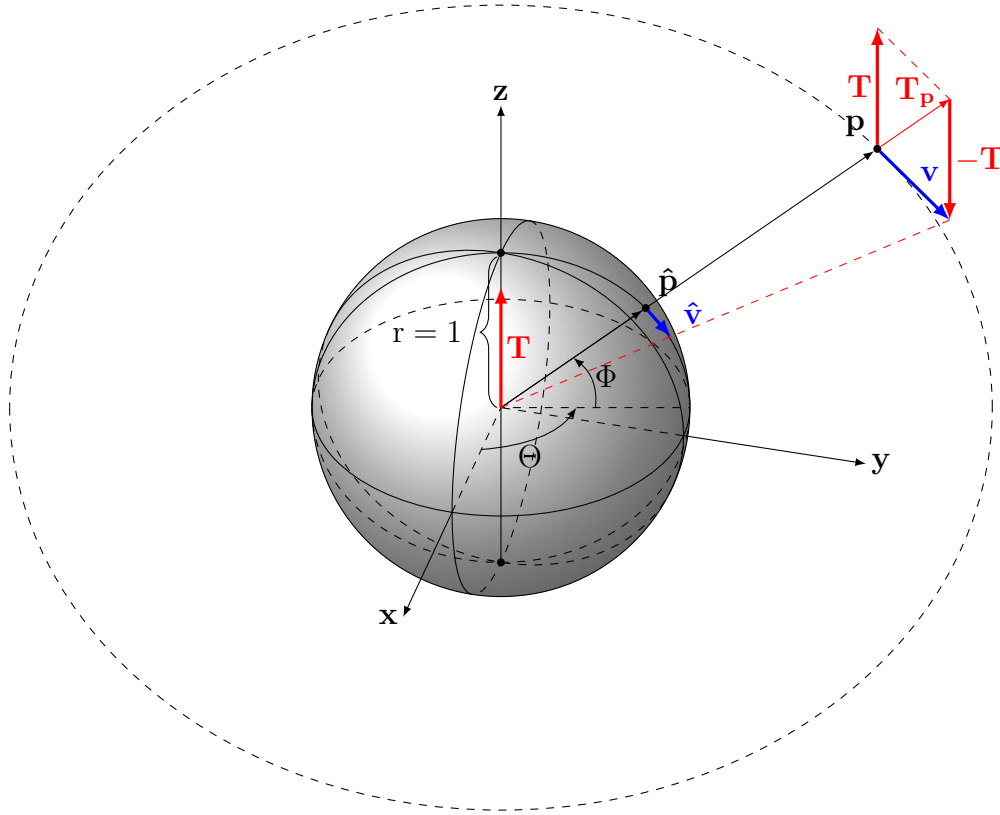


Figure 2.3: The translation \mathbf{T} of a spherical sensor (gray sphere) causes a relative motion of the point \mathbf{p} in the sensor's environment that leads to the movement of the point's projection $\hat{\mathbf{p}}$ on the sensor's surface. The optical flow vector $\hat{\mathbf{v}}$ describes the velocity of the projected point $\hat{\mathbf{p}}$.

The optical flow vector $\hat{\mathbf{v}}$ that describes the velocity of the projected point $\hat{\mathbf{p}}$ can be derived from the original point's position \mathbf{p} , the sensor's translation \mathbf{T} , and the velocity component of the point \mathbf{p} that is perpendicular to the sensor's surface. Since both velocity vectors $\hat{\mathbf{v}}$ and \mathbf{v} are perpendicular to $\hat{\mathbf{p}}$ and \mathbf{p} , one can derive \mathbf{v} from the projection of \mathbf{T} onto $\hat{\mathbf{p}}$ that will be called \mathbf{T}_p :

$$\mathbf{v} = \mathbf{T}_p - \mathbf{T} \quad (2.4)$$

To derive an analytical description of the optical flow vector field $\hat{\mathbf{v}}$, one should first recall the equations for the projection of vectors. The projection of a vector \mathbf{a} onto a vector \mathbf{b} is shown in figure 2.4. The resulting vector \mathbf{c} can be found as follows: From geometry it is known that

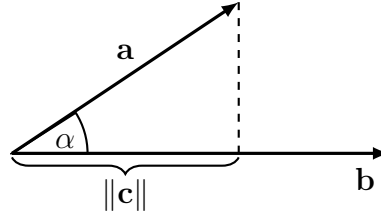


Figure 2.4: Projection of a vector \mathbf{a} onto another vector \mathbf{b} . The resulting vector \mathbf{c} points in the same direction as \mathbf{b} but has the norm $\|\mathbf{c}\|$. α is the angle between the vectors \mathbf{a} and \mathbf{b} .

$$\cos(\alpha) = \frac{\|\mathbf{c}\|}{\|\mathbf{a}\|} \quad (2.5)$$

The definition of the scalar or dot product of \mathbf{a} and \mathbf{b} is

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\| \|\mathbf{b}\| \cos(\alpha) \quad (2.6)$$

By combining equations (2.5) and (2.6) one can find the norm $\|\mathbf{c}\|$ of vector \mathbf{c}

$$\begin{aligned} \mathbf{a} \cdot \mathbf{b} &= \|\mathbf{a}\| \|\mathbf{b}\| \frac{\|\mathbf{c}\|}{\|\mathbf{a}\|} \\ \|\mathbf{c}\| &= \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{b}\|} \end{aligned} \quad (2.7)$$

Vector \mathbf{c} is received by multiplying its norm with a unity vector pointing in the direction of vector \mathbf{b} :

$$\begin{aligned} \mathbf{c} &= \|\mathbf{c}\| \frac{\mathbf{b}}{\|\mathbf{b}\|} \\ \mathbf{c} &= \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{b}\|^2} \mathbf{b} \end{aligned} \quad (2.8)$$

Combining (2.8) and (2.4) and taking into account that $\|\hat{\mathbf{p}}\| = r = 1$, \mathbf{T}_p can be replaced

$$\mathbf{v} = (\mathbf{T} \cdot \hat{\mathbf{p}}) \hat{\mathbf{p}} - \mathbf{T} \quad (2.9)$$

and a general description of the optical flow vector $\hat{\mathbf{v}}$ that is caused purely by translation of the spherical sensor can be derived:

2.2. Formal description of optical flow perceived by a perfect spherical sensor¹⁷

$$\hat{\mathbf{v}} = \frac{1}{\|\mathbf{p}\|} [(\mathbf{T} \cdot \hat{\mathbf{p}}) \hat{\mathbf{p}} - \mathbf{T}] \quad (2.10)$$

2.2.2 Optical flow induced by sensor's rotation

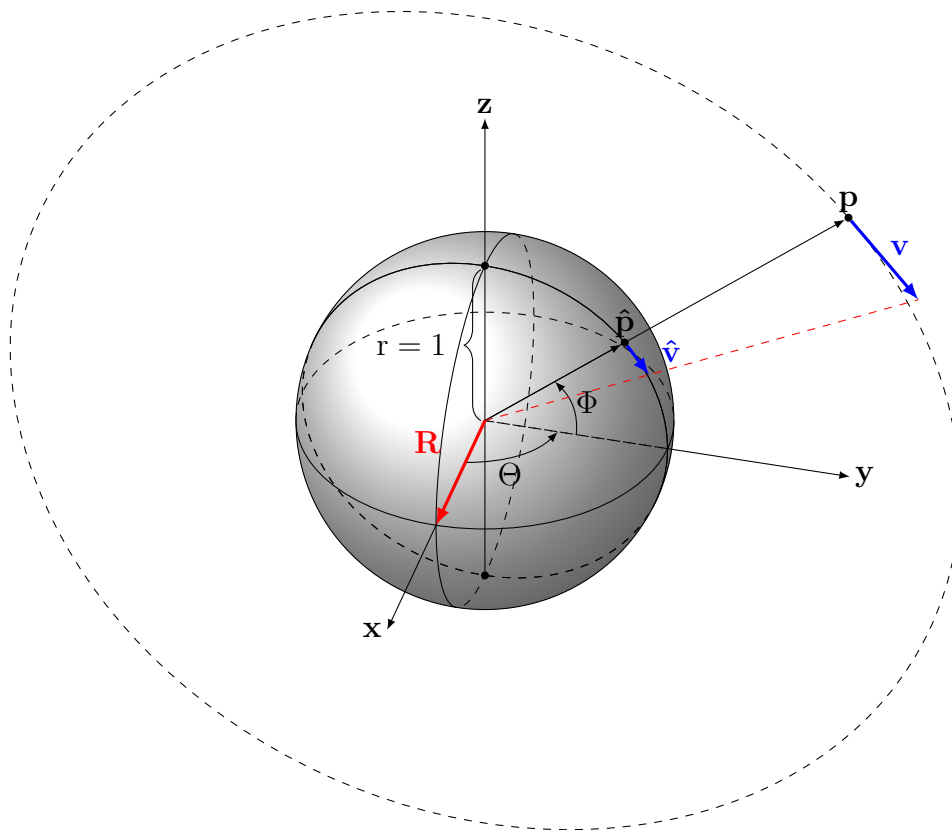


Figure 2.5: The rotation \mathbf{R} of a spherical sensor (gray sphere) causes a relative motion of the point \mathbf{p} in the sensor's environment with an angular velocity \mathbf{v} . The optical flow vector $\hat{\mathbf{v}}$ describes the velocity of the projection of the point $\hat{\mathbf{p}}$ on the sensor's surface.

The rotation of a spherical sensor can be described in polar coordinates by the two angles azimuth Θ and elevation Φ or in Cartesian coordinates by the rotation vector \mathbf{R} that is orthogonal to the plane that describes the sensor's rotation (figure 2.5). The velocity or optical flow vector $\hat{\mathbf{v}}$ of the projection $\hat{\mathbf{p}}$ of any arbitrary point \mathbf{p} in the sensor's environment is then given by

$$\hat{\mathbf{v}} = -\mathbf{R} \times \hat{\mathbf{p}} \quad (2.11)$$

2.3 Conclusions: theoretical properties and limitations of optical flow

In conclusion to the derivation of the equations (2.10) and (2.11) several basic observations or facts on optical flow can be formulated:

Observation 1: The optical flow caused by self-motion of a sensor that is both translating and rotating is a **linear combination of the optical flow fields** that would have been caused by separate translation and rotation. Thus (2.10) and (2.11) can be combined to (Koenderink and van Doorn, 1987):

$$\hat{\mathbf{v}} = \frac{1}{\|\mathbf{p}\|} [(\mathbf{T} \cdot \hat{\mathbf{p}}) \hat{\mathbf{p}} - \mathbf{T}] - \mathbf{R} \times \hat{\mathbf{p}} \quad (2.12)$$

From (2.12) the optical flow fields $\hat{\mathbf{v}}(\Theta, \Phi)$ that a spherical sensor perceives during translation (figure 2.6) and rotation (figure 2.7) using only one of its possible six degrees of freedom (DOFs) at a time can be generated. The left side of the figures 2.6 and 2.7 shows the optical flow vectors being directly projected onto the sensor's surface while the figures' right sides show the corresponding Mercator projections of the spherical optical flow fields. The gray line in the plots marks the equator and the red arrows indicate the direction of egomotion. The gray arrow in the spherical plots indicates the direction of forwards motion and corresponds to the center position in the Mercator projections. Due to the principle of linear combination, optical flow fields perceived by a sensor translating in different directions and/or rotating around several axis of rotation at the same time can be found by linear combination of the optical flow fields caused by self-motion using the different types of motion separately.

Observation 2: From (2.12) it can be concluded that **only optical flow caused by translation of the observer is dependent on the distance to the**

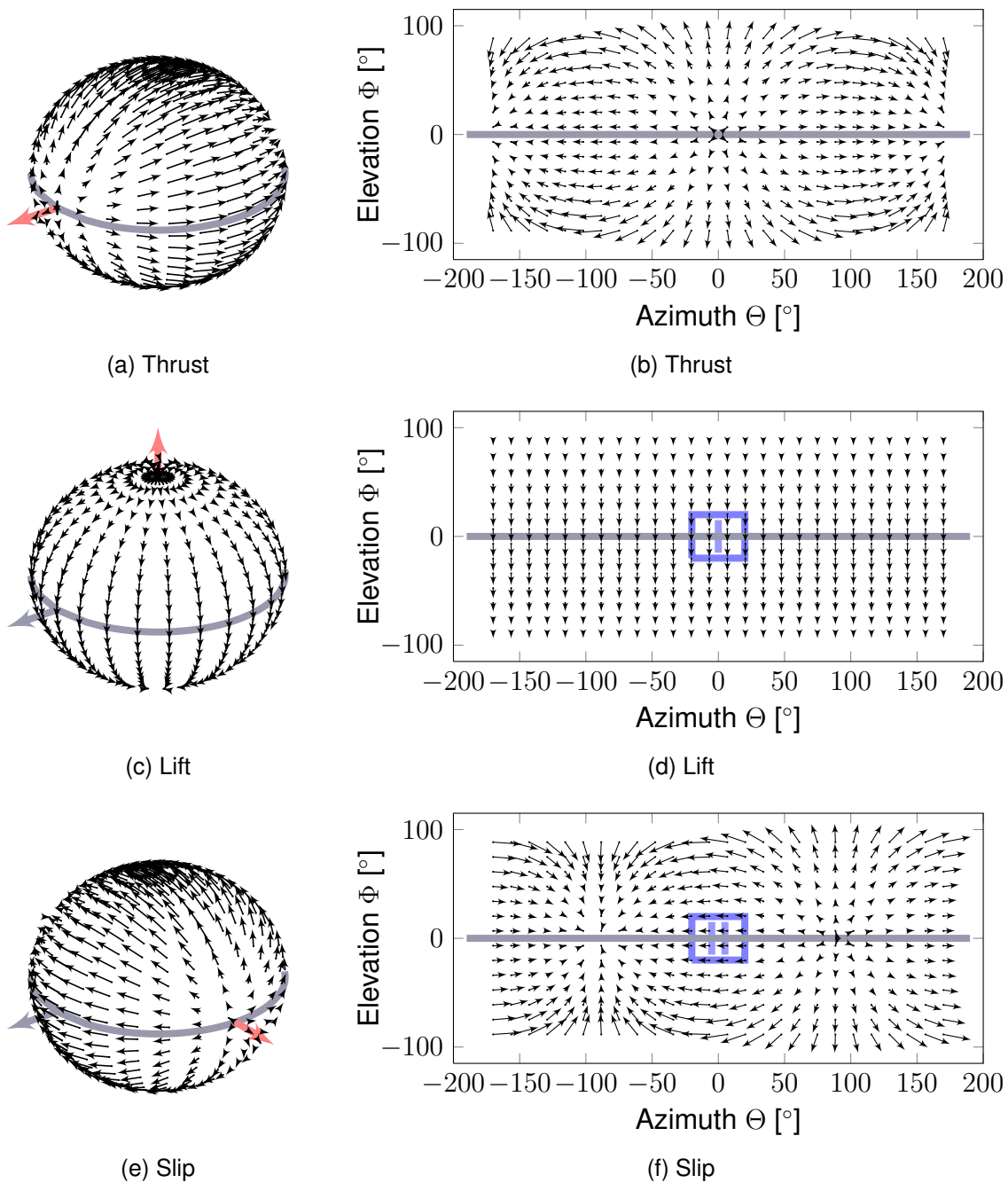


Figure 2.6: (left) Simulated 3D optical flow fields that would be measured by spherical optical flow sensors and (right) their corresponding 2D projections onto Mercator maps for different types of egomotion of the sensor. The gray line corresponds to the equator line. The gray arrow indicates the forward direction. The position of this arrow corresponds to the center of the Mercator maps. The red arrow indicates the direction of translation of the sensor. Optical flow fields as shown that are caused by translation due to (a),(b) thrust, (c),(d) lift, and (e),(f) slip.

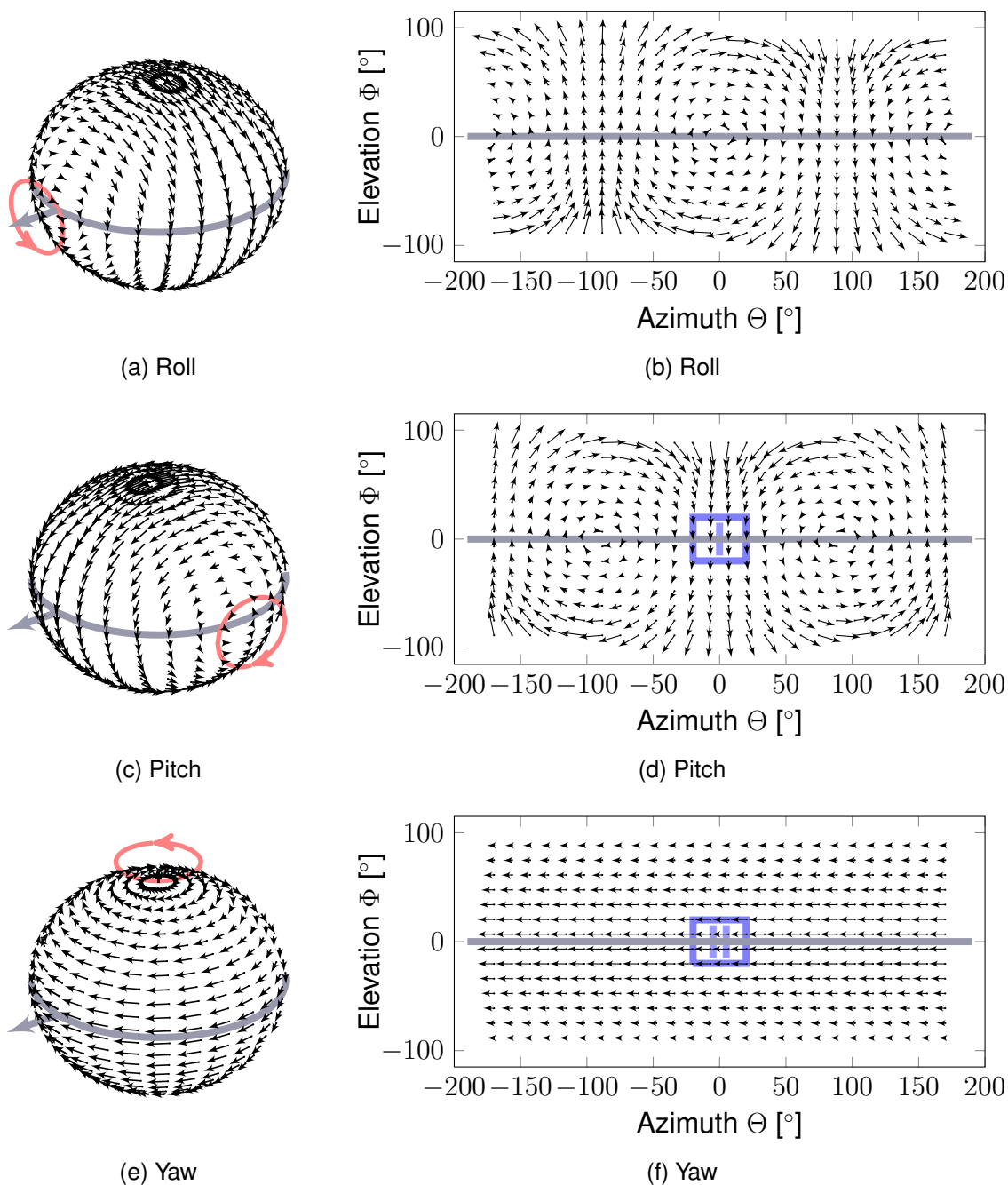


Figure 2.7: (left) Simulated 3D optical flow fields that would be measured by spherical optical flow sensors and (right) their corresponding 2D projections onto Mercator maps for different types of egomotion of the sensor. The gray line corresponds to the equator line. The gray arrow indicates the forward direction. The position of this arrow corresponds to the center of the Mercator maps. The red arrow indicates the direction of rotation of the sensor. Optical flow fields as shown that are caused by rotation around the (a),(b) roll, (c),(d) pitch, and (e),(f) yaw axis.

objects in the observer's environment. The amplitude of optical flow is inversely proportional to the distance between the observer and the objects. This means that the relative distance of objects that are closer to the observer can be more easily distinguished than the distance of objects that are further away. For non-perfect optical flow sensors with limited resolution, there will be a distance from which all objects will appear to be moving with zero velocity. In flight control this is typically not of concern since these objects will be considered to be too far away to be taken into account for the control of the flying platform.

Observation 3: Without additional information **translational optical flow can only provide relative distance information**. For instance the optical flow on the left and right sides can be balanced for an observer to pass through the middle of a tunnel but without any additional knowledge about the velocity of the observer, the distance or the size of a known object in the environment, an estimation of the absolute distance to the walls is not possible. As described in sections 3.2 and 3.1 relative distance information extracted from optical flow is used by insects like bees. By regulating optical flow, bees for example automatically decelerate when approaching the floor or when entering narrow tunnels where there is less space to maneuver.

Observation 4: Even during pure translation of the observer, objects with the same distance from the observer will cause different amounts of **optical flow depending on the objects' positions relative to the forward movement** of the observer. This can be best seen from figure 2.6b. This plot was generated assuming that the observer is moving "forward" - so the center of the Mercator map corresponds to the direction in which the observer is moving. All objects are assumed to have the same distance from the observer. Under these circumstances a well-known expansion pattern is detected by an optical flow sensor where the optical flow in the center of the pattern is zero. As a result if an observer is constantly moving forward towards a small object, this object cannot be detected using optical flow and the observer would directly run into the object. In the control of artificial MAVs this problem is typically solved by simply not allowing any small objects like wires or sticks in the environment of the MAV.

Observation 5: A well-known fact and a result of (2.12) is that from image expansion

one can estimate the **time to contact** a surface (Srinivasan and Zhang, 2004). When an observer is flying normal to a surface as shown in figure 2.8 the time to contact the surface t_{ttc} when the observer approaches with a constant speed is given by

$$t_{ttc} = \frac{\Theta}{\dot{\Theta}} \quad (2.13)$$

where Θ is the angle between the vector pointing in the direction of motion of the observer and the vector pointing towards a point X on the surface and $\dot{\Theta}$ is the rate giving how quickly the angle Θ changes per unit time.

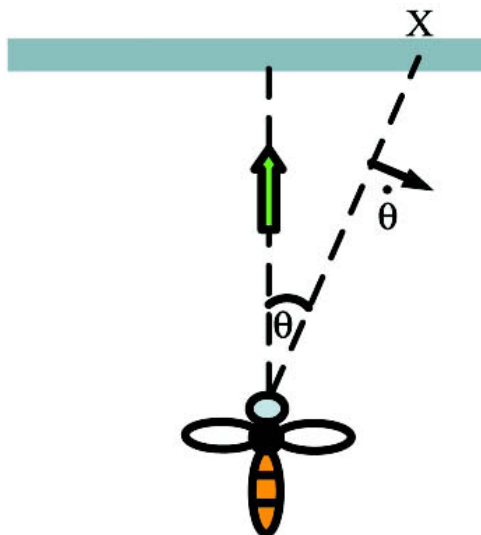


Figure 2.8: When an observer like a bee is flying towards a surface, it can estimate its time to contact from the ratio between angle Θ of a feature X and the temporal derivative of that angle $\dot{\Theta}$ given that the observer continues moving with the same speed. Figure is adapted from Srinivasan and Zhang (2004).

Observation 6: The **information contained by local patches** of the global optical flow fields is typically **ambiguous**. See the patches "I" in the figures 2.6d and 2.7d and the patches "II" in figures 2.6f and 2.7f. Although each pair of patches belongs to optical flow fields caused by different types of egomotion of the observer, the local patches look very similar and cannot be distinguished when perceived by a non-perfect optical flow sensor with limited spatial resolution. The optical flow information from

other patches or sensors is needed to precisely estimate the observer's self-motion.

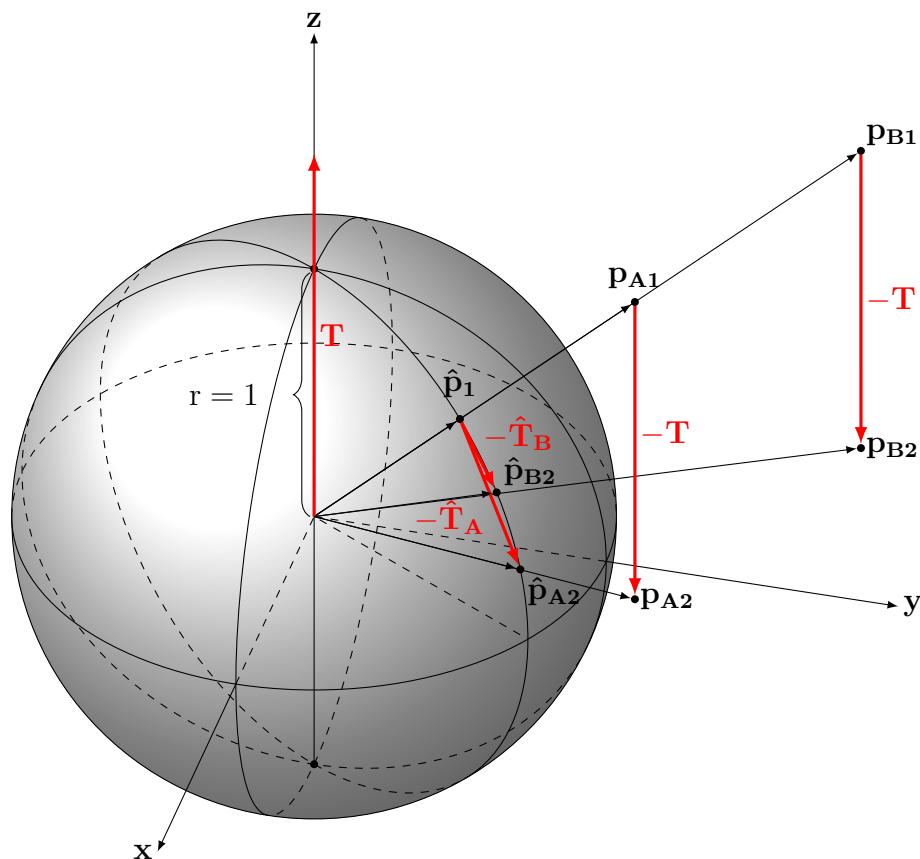


Figure 2.9: Optical flow caused by translation of an observer is dependent on the distance between the observer and a point-like object whose projection moves on the observer's sensor surface (depicted as gray sphere). At time 1, two points p_{A1} and p_{B1} are placed so that p_{A1} hides p_{B1} and the projection of both points onto the observer's sensor surface is \hat{p}_1 . However, due to the different distances between the points and the observer their projections travel a different distance on the observer's sensor surface when the observer translates by the vector T . As a result after the translation at time 2, p_{A2} is not occluded by p_{B2} and each point has its own projection \hat{p}_{A2} and \hat{p}_{B2} . So p_B becomes visible in the observer's field of view.

Observation 7: The fact that distant objects seem to move more slowly than close objects leads to the observation that **objects can appear and disappear if the observer translates relative to the objects**. Figure 2.9 depicts two points p_A and p_B in the environment of a spherical sensor (shown

as gray sphere). If the sensor moves along the translation vector \mathbf{T} both points are shifted relative to the sensor by the vector $-\mathbf{T}$ from position 1 to position 2 and their projections on the spherical sensor change from the same initial position \hat{p}_1 to the different positions \hat{p}_{A2} and \hat{p}_{B2} . So while p_B was initially occluded by p_A it becomes visible to the sensor after the translation. Since the translation can be reversed after the observer moves by $-\mathbf{T}$, p_B would again be hidden behind p_A .

Observation 8: In general, the projection of a 3-dimensional world on the 2-dimensional surface of a sensor causes **optical flow fields** that are **ambiguous**. Their interpretation without prior knowledge or assumptions can be difficult. If the observer operates in an environment where objects are free to move the very same optical flow fields can be generated by the observer's self-motion or by the movement of objects. Without additional knowledge, it is impossible to distinguish with absolute certainty if the optical flow pattern was caused by self-motion or motion of the environment. In addition, zero optical flow does not mean that the observer is stationary but could simply mean that the objects in its environment compensate for the optical flow field by moving at the same speed in the opposite direction. This phenomena is widely exploited by experimentalists in biology (David (1982); Srinivasan et al. (1991); Krapp and Hengstenberg (1996); Krapp et al. (1998); Fry et al. (2009)) to study the behavior of insects caused by perceived optical flow. To overcome this problem in the control of aerial vehicles, it is often assumed that other objects are not moving or at least sufficiently slowly so that they cannot cause any harm.

From these observations it can be concluded that as stated by Gibson (1950) the optical flow generated during self-motion of the observer is indeed a rich source of information about the observer's self-motion and the objects in the scene, but the interpretation of the observed optic flow field patterns is not trivial. In the presence of noise local optical flow patches cannot provide unambiguous information about the observer's self-motion and the same optical flow can be generated either by the observer's self-motion or by moving objects in the observer's environment. In the following chapter 3 I present studies on flight control in flies and honeybees to provide a better understanding on how these animals extract and use optical flow for behavioral guidance.

Chapter 3

Optical flow based control in flies and honeybees

Abstract

This review chapter analyzes how optical flow is used in nature for guiding the behavior of flying insects like flies and honeybees. There is a large amount of literature about flight control and vision systems of insects as well as on the vision systems of higher vertebrates. However, it is still not completely understood how insects' brains, vision systems and bodies allow for their amazing flight skills. From the large amount of information in the literature this chapter concentrates on the following well-known facts about the vision system and flight control of insects: (1) Flies and bees regulate flight speed with optical flow. (2) Honeybees measure flight distances with optical flow. (3) Insects use compound eyes that are optimized for the perception of optical flow. (4) Visual interneurons in blowflies estimate self-motion. From the examples it can be concluded that flies and honeybees developed several strategies to extract information about self-motion and the environment using optical flow and to deal with possible imperfections in the extracted optical flow field.

3.1 Flies and bees regulate flight speed and wall distance through optical flow

There is an extensive literature that describes how flies and honeybees regulate their flight behavior based on optical flow (Srinivasan et al., 2000b; Srinivasan and Zhang,

2004; Tammero and Dickinson, 2002*a,b*; David, 1982; Fry et al., 2009). This section presents two experiments by Srinivasan et al. (1991, 1996) which describe the behavior of bees when flying through tunnels.

In the **first set of experiments** (figure 3.1) Srinivasan et al. (1991) studied the flight behavior of honeybees when flying in tunnels. The flanking walls of the 40 cm long, 12 cm wide, and 20 cm high tunnel (figure 3.1a) were **covered with stationary and moving black-and-white gratings**. From recordings of more than hundred of flight trajectories of bees, Srinivasan et al. found that bees tend to fly in the middle of the tunnel when both walls are stationary (figure 3.1b), closer to walls where the grating moved in the same direction as the bee (figure 3.1c), and further away from gratings moving in the opposite direction (figure 3.1d). Srinivasan et al. concluded that bees do not measure the absolute distance of the walls but instead balance the amount of optical flow perceived by their left and right eyes (Srinivasan et al., 1991; Srinivasan and Zhang, 2004). Furthermore, similarly to the results published by Si et al. (2003), Srinivasan et al. found that this optical flow balancing mechanism works robustly independent of contrast and spatial frequency of the tunnel gratings.

In 1996 Srinivasan et al. presented a **second set of experiments on speed regulation of honeybees in narrow tunnels** (Srinivasan et al., 1996). Srinivasan et al. trained honeybees to fly through a tunnel covered with stationary black-and-white gratings and variable distance between its walls (figure 3.2). Figure 3.2b shows a top view of the tunnel together with a typical flight trajectory of a bee. The trajectory indicates that similarly to the results from the first set of experiments by Srinivasan et al. (1991) bees try to fly approximately in the middle of the tunnel. Figure 3.2c presents a speed profile from 18 flights in the tunnel. Error bars indicate the mean and standard deviation of flight speeds at different positions in the tunnel. The speed profiles indicate that bees try to control their own speed to perceive a constant amount of optical flow by both eyes corresponding to an angular velocity of images perceived from the walls of $320^\circ/\text{s}$ (dashed line in figure 3.2c). Using this mechanism, bees slow down when the distance between walls becomes smaller, and accelerate when the distance increases.

3.2 Honeybees use optical flow to measure flight distances

Honeybees can perform nearly acrobatic flight movements and demonstrate impressive landing skills when for instance landing on flowers moving in the wind and due to the

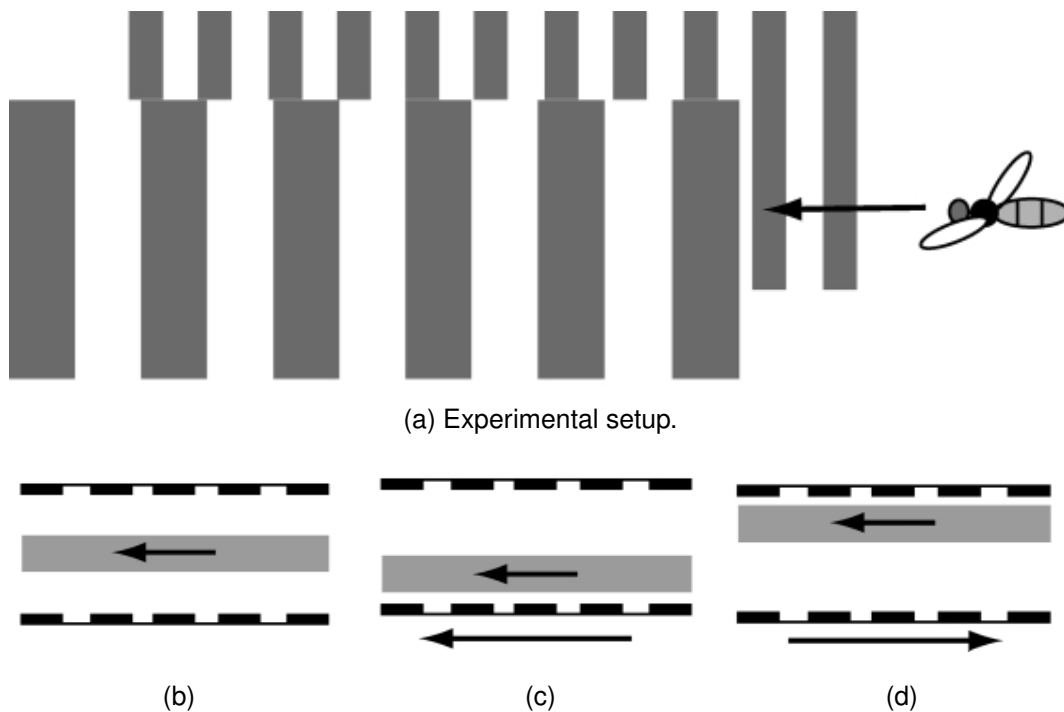


Figure 3.1: Figures (b) - (d) show the mean and standard deviation (gray areas) of more than 100 flight trajectories of bees flying through a tunnel with black-and-white gratings (figure a). Small arrows indicate the flight direction of bees. Big arrows show the direction of pattern movement. Bees tend to fly in the middle of the tunnel when the gratings on the walls are not moving (b), closer to gratings moving in the same direction (c), and further away from gratings moving in the opposite direction (d). Figures (a) - (d) are adapted from Srinivasan et al. (1991).

bees weight. To explore new food sources, bees travel several kilometers. After returning to the hive, bees perform a waggle dance (Von Frisch, 1967) with which they share their knowledge about food sources with other bees - giving precise instructions about how to reach a food source.

Srinivasan et al. (2000a) have been studying the question of how honeybees measure the distance to a food source. At the time of their experiments there had been two major theories: (1) Theory 1 stated that bees might measure of how much energy they use during their flight to the food source (Heran, 1956). Communicating this energy measurement to other bees might be useful since it encodes both the distance traveled and allows other bees to estimate if the flight is worth given the amount of food that can be found and brought back to the hive. However, when wind conditions change, the amount of energy spent on a flight varies and thus leads to a unreliable distance

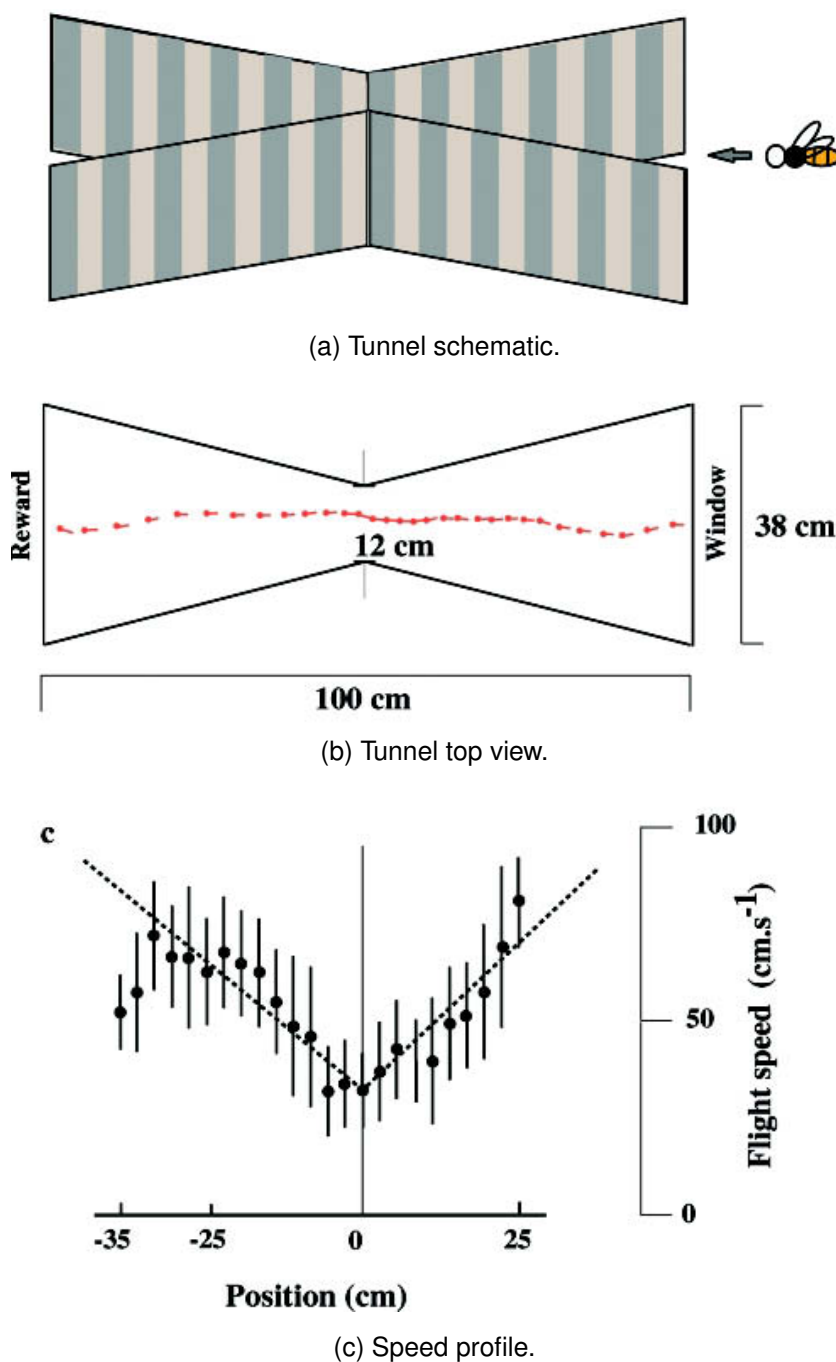


Figure 3.2: Recordings of flight velocities of honeybees (c) flying through a tunnel with black-and-white gratings (a). The tunnel has its narrow point in the middle (see the tunnel's top view in (b)). The dashed line in (b) presents a typical flight trajectory. Error bars in (c) show the mean and standard deviation of flight speeds at different positions in the tunnel (data from 18 flights). The measurements indicate that bees automatically slow down when the distance between tunnel walls decreases and accelerate again once the tunnel becomes wider again. The dashed line in (c) represents the theoretically predicted speed profile if bees try to receive a constant optical flow with an angular velocity of $320^\circ/\text{s}$. Figures (a) - (c) are adapted from Srinivasan et al. (1996).

estimate. (2) Theory 2 which was supported by the results found by Srinivasan et al. (2000a) stated that bees measure the accumulated amount of optical flow perceived during the flight which they also communicate during the waggle dance:

To evaluate the role of optical flow in the bees' perception of the distance between a hive and a food source, Srinivasan et al. trained bees to collect from a food source that they placed in an artificial tunnel whose walls are covered with patterns of different texture. Four experiments have been made as depicted in figure 3.3a:

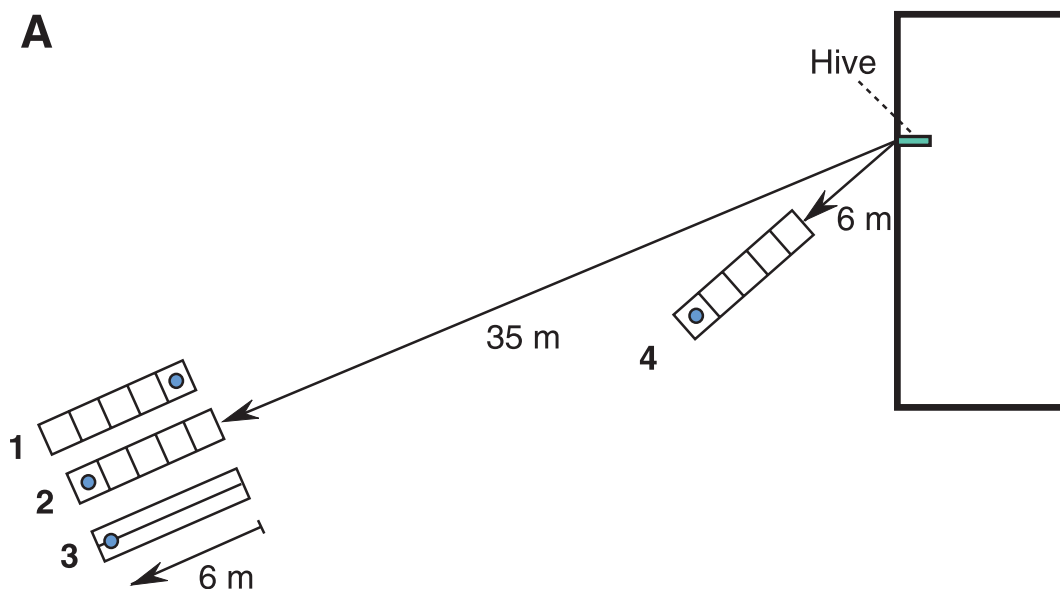
Experiment 1: A food source was placed at the entrance of an artificial tunnel whose walls were covered with a random visual texture. The tunnel was placed 35 meters away from the hive.

Experiment 2: The food source was placed in the same tunnel as in experiment 1 but this time at the end of the tunnel. The tunnel was closed at its end so that bees had to travel 6 meters inside the tunnel to reach the food source.

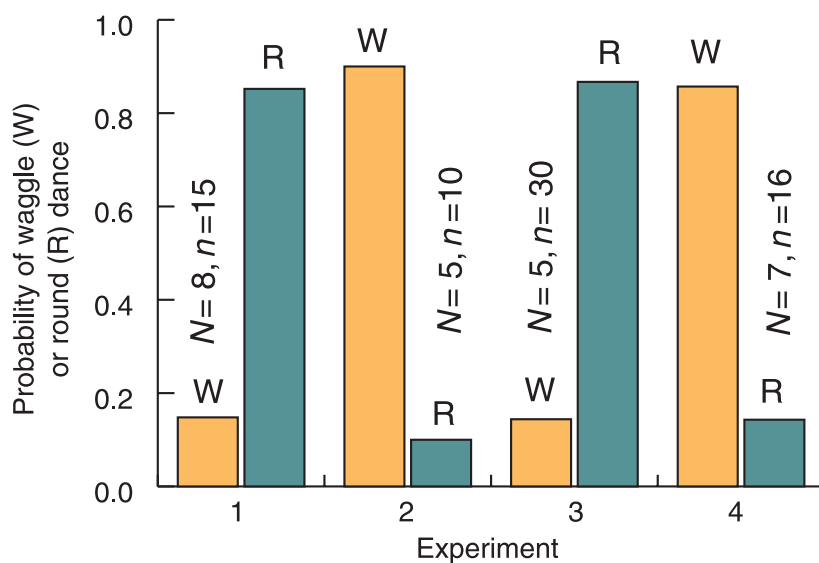
Experiment 3: This experiment had the same conditions as in experiment 2 except that the tunnel wall was covered with a pattern texture that did not lead to any high optical flow measured in the bees' visual system.

Experiment 4: The same conditions as in experiment 2 were chosen except that the tunnel was placed only 6 meters away from the hive.

Srinivasan et al. found that under the conditions of the first and third experiments, more than 85% of the bees performed round dances (see figure 3.3b). It is known from Boch (1957) that the honeybees (*Apis mellifica* L.) used in the experiments typically perform round dances when the food sources are within the range of 50 meters of the hive - otherwise the bees prefer to perform a waggle dance. Interestingly more than 85% of the bees performed such a waggle dance under the conditions of experiments two and four although in both experiments the food source was clearly a distance below 50 meters from the hive. Srinivasan et al. concluded that the narrow tunnel in experiments two and four caused bees to perceive a large amount of optical flow which lead to an overestimate of the actual distance between the hive and the food source. This overestimate lead them to change their style of dance. This theory was supported by the fact that the bees mostly did not perform a waggle dance in experiment 3 although they were forced to travel the same distance along the same tunnel as in experiment 2 except for the fact that the different texture on the walls of the tunnel in experiment 3 did not cause a large optical flow.



(a) Experimental setup.



(b) Results

Figure 3.3: (a) For experiments 1-4, food sources indicated as blue dots are placed at different positions in a tunnel. (b) The results indicate that the probability of a bee performing a waggle (W) or a round (R) dance varies with the amount of optical flow that the bee perceived during the flight. N and n respectively represent the numbers of bees and dances that have been analyzed. Figures (a) and (b) are taken from Srinivasan et al. (2000a).

In a second set of experiments, Srinivasan et al. found that for bees traveling a distance over 50 meters, the duration of their waggle dance is proportional to the amount of optical flow the bees perceived during their flight.

3.3 Insects' compound eyes are optimized for perception of optical flow

The eyes of most insects are different from those of human beings. Humans have two mobile eyes each of them with a single lens with controllable focal length and a retina with several millions of photoreceptors. The strong overlap between the visual fields of the two eyes supports the stereo vision that vertebrates are known to use for distance estimation of surrounding objects (Srinivasan and Zhang, 2004).

Since stereo vision requires a sufficient distance between two eyes, this mechanism cannot be exploited by small insects for a reliable estimation of distances (Rossel, 1983; Horridge, 1987). Instead insects evolved a vision system based on an array of a few thousand facets or ommatidia that form compound eyes like those shown in figure 3.4. Each ommatidium contains a fixed-focus lens with a field of view of a few degrees that focuses light on a small group of photoreceptors (Srinivasan and Zhang, 2004). The arrangement of ommatidia allows no big overlap of receptive fields but instead nearly panoramic vision (Wehner, 1981) supporting the perception of panoramic optical flow.

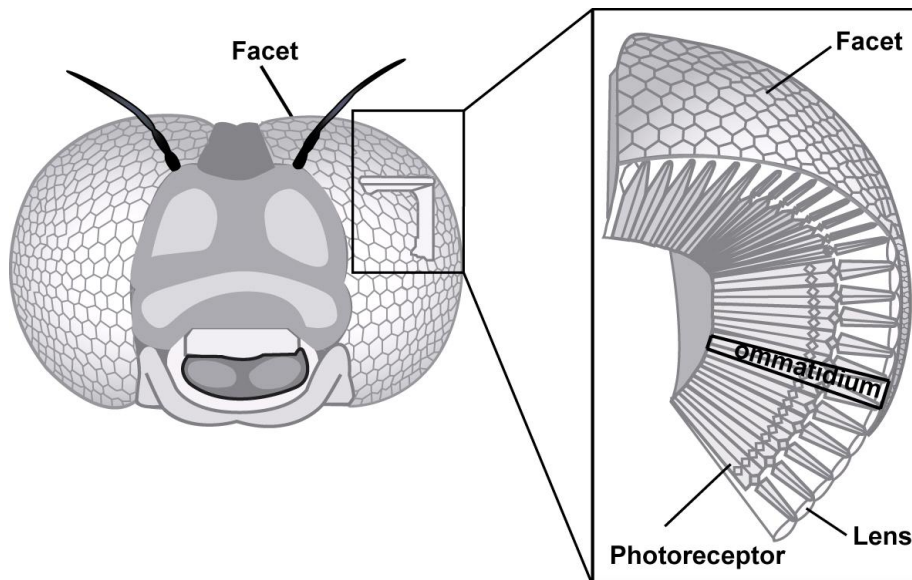
3.4 Visual interneurons in blowflies estimate self-motion

Since it is known that insects' flight control might be based on optical flow, researchers have studied the responses of motion-sensitive neurons in insects (Hausen, 1993; Krapp and Hengstenberg, 1996; Krapp, 1999). Reviews have for instance been published by Hausen and Egelhaaf (1989); Hausen (1993) and Egelhaaf and Borst (1993b). This section describes two selected publications by Krapp and Hengstenberg (1996) and Krapp et al. (1998) who found that the visual interneurons of flies estimate the self-motion of the fly by using optical flow.

It is known that insects contain large arrays of elementary motion detectors (EMDs) each with a small field of view and a preferred direction of motion (Reichardt, 1987; Krapp et al., 1998). Krapp and Hengstenberg (1996) and Krapp et al. (1998) studied interneurons in blowflies with large receptive fields that are known to directly or indi-



(a) Picture of compound eye.



(b) Schematic of compound eye

Figure 3.4: (a) Picture of the head and compound eyes of a hover-fly (*Volucella pellucens*) taken from Bartz (2011). (b) Schematic of an insect's compound eye taken from LIS (2011). Several thousands of ommatidia or facets form a compound eye. Each ommatidium contains a lens that focuses light on a small number of photoreceptors.

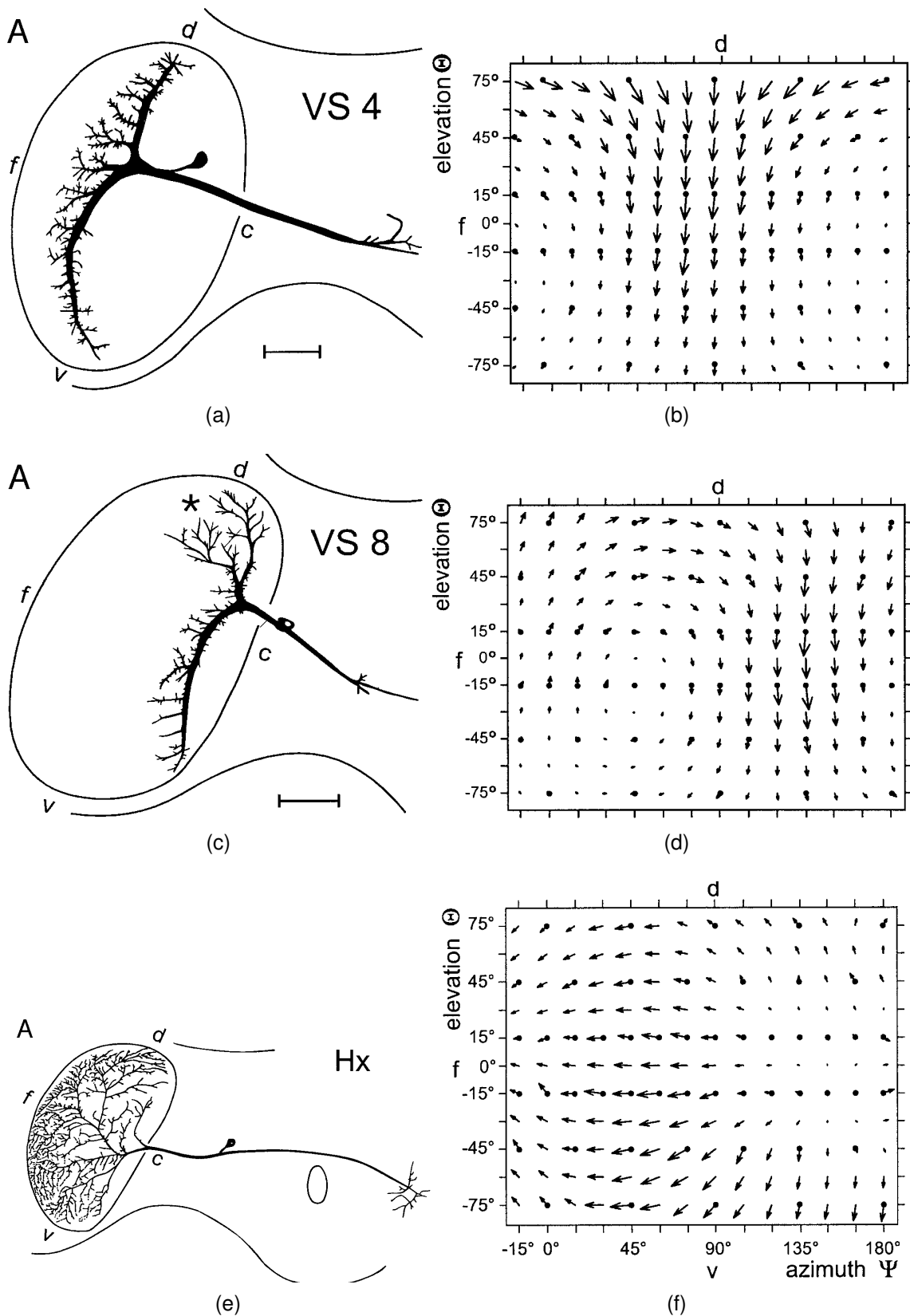


Figure 3.5: Anatomy (a,c,e) and optical flow response fields (b,d,f) of interneurons in blowflies. Figures (a) - (f) are adapted from Krapp et al. (1998).

rectly receive input from these large arrays of EMDs. By mapping the receptive field of these interneurons, and presenting complex visual motion stimuli while recording intracellularly from the cells they found that some of the interneurons' motion response fields were remarkably similar to optical flow fields. During their evaluation of so-called giant vertical cells (or VS cells) (Hengstenberg, 1982; Hengstenberg et al., 1982) Krapp and Hengstenberg (1996) and Krapp et al. (1998) found that some neurons preferred pitch-rotation (VS1-VS3) while a second and third group of neurons was tuned to roll-rotation (VS4-VS7) and a mix of pitch and roll rotation (VS8-VS10), respectively. So by reading the signals of all VS neurons a reliable measure of body rotation is provided. Furthermore, Krapp and Hengstenberg (1996) found a neuron that they labeled Hx which prefers stimuli similar to optical flow fields generated during translation of the fly. Pictures of the neurons as well as their optical flow response fields are shown in figure 3.5. The response fields on the right side of figure 3.5 look very similar to the theoretically generated optical flow fields shown in figures 2.6 and 2.7 in section 2.3

3.5 Conclusions

Although in the scope of this thesis it is not possible to analyze all material on insects' vision systems and flight control this chapter presents several facts that indicate that insects like flies and honeybees have evolved control and vision systems that exploit optical flow for behavioral guidance and flight control. Optical flow is a main stimulus for controlling behavior in insects like flies and honeybees that actively use optical flow for flight control and distance estimation (Srinivasan and Zhang, 2004). The knowledge about how insects exploit optical flow and how their vision and control systems as well as their body properties have been adapted to gain maximum information from optical flow is crucial for understanding how to build better visually guided micro aerial vehicles and optical flow sensors.

It is further important to understand how insects deal with the imperfect information that - as predicted from the perfect optical flow sensor model presented in the previous chapter 2.3 - can only be extracted from the optical flow field as well as with imperfections in their vision system that has to deal with limited stimulus contrast and noise.

- The two sets of experiments presented by Srinivasan et al. (1991, 1996) show how honeybees use optical flow for steering close to obstacles but also how valuable optical flow can be for the control of flying vehicles. The fact that optical flow does not allow for an absolute measure of distance (observation 3, section 2.3)

does not need to be a disadvantage. The behavior of bees proves that a measure like optical flow that directly relates flight speed and distance of objects in the environment indeed can be of benefit. Optical flow allows for automatically slowing down when stationary objects come closer thus the space for maneuvering is decreased. This is a desirable behavior for MAVs. Balancing the optical flow perceived on the left and the right sides when passing through a tunnel does not only allow to keep the position of the MAVs in the middle of a tunnel but as well makes them avoid objects moving towards the MAVs that are typically a greater threat and risk for collision than objects moving with or away from the MAV. So in summary the lack of an absolute distance measure might require a rethinking of the control strategy for MAVs but in the end can prove to be an advantage rather than a problem.

- The experimental results presented by Srinivasan et al. (2000a) show that based on optical flow honeybees are capable of providing an absolute distance measure between the hive and a food source. Accumulating the amount of optical flow perceived during a flight gives a distance measure that is independent from flight speed. Communicating the amount of optical flow perceived during a flight as a measure of a distance rather than the amount of energy spend during that flight is of advantage. Conditions like the direction and amount of wind can change rapidly and often during and in between flights affecting the measure of the energy being spend for a flight. So wind effects would make a distance measure based on consumed energy noisy and unreliable. However, wind does not affect the amount of optical flow perceived during the flight. It is very unlikely that the environment changes so rapidly that the optical flow measurements become unreliable.

An phenomena that is not fully understood yet is how bees deal with different altitudes during the flight. As shown in observation 2 (section 2.3) the absolute magnitude of translational optical flow is inversely proportional to the distance to the objects. So the amount of optical flow perceived during the flight by a bee depends on its altitude (Esch and Burns, 1995, 1996) and on the distance to vertical surfaces (Srinivasan et al., 1996, 1997; Si et al., 2003). Indeed studies show that when bees are forced to fly at a high altitude these bees perform a waggle dance with a shorter mean duration indicating a lower amount of optical flow being received during the flight (Esch and Burns, 1996). The key to an absolute distance measure might be that bees try to keep the average optical flow perceived during a given amount of time constant and thus automatically select a high altitude when flying fast and slowing down when flying close to the ground or a vertical surface

- similarly as bees slow down in narrow tunnels. In any case the fact that bees only communicate the amount of perceived optical flow and direction to a food source during their waggle dance and not the precise nature of the environment between the hive and food source (Esch et al., 2001) indicates that bees must have some standard behavior that allows for reliable distance estimation based on optical flow.
- In the presence of noise local optical flow patches cannot provide unambiguous information about the observer's self-motion (observation 6, section 2.3). To overcome these ambiguities in local optical flow patches insects evolved a panoramic vision system. The results by Krapp and Hengstenberg (1996) and Krapp et al. (1998) present evidence that insects spatially integrate the local optical flow signals gained from local elementary motion detectors and use global optical flow to estimate self-motion on the neuronal level. This spatial integration allows not only to resolve locally ambiguous optical flow information but also to reduce noise. So a similar set of global optical flow filters could as well be useful for the control of MAVs.
 - Despite the ambiguities in optical flow interpretation insects like bees and flies demonstrate successful flight control, landing skills and escape from enemies by fusing the optical flow perceived through their compound eyes as the main source of information for behavioral control with additional information for instance from relative airflow and inertia sensors (Srinivasan and Zhang, 2004; Taylor and Krapp, 2007). It is known that course control and stabilization is supported by halteres. Halteres are small hind-wings that oscillate in antiphase with the main wings (Dickinson, 1999; Nalbach, 1993; Nalbach and Hengstenberg, 1994; Srinivasan and Zhang, 2004). They measure forces normal to their plane of oscillation that are generated during turns. But it is not entirely clear how they feed back into the fly's control system. From an excessive study of the existing literature Taylor and Krapp (2007) concluded that "it appears that halteres do measure angular velocity, but as an emergent property of the control system of which they are a part." So one can imagine that halteres act like miniature rate gyroscopes that measure the angular velocity of the fly's body (Srinivasan and Zhang, 2004). There is evidence that halteres are used to compensate for rapid rotations while optomotor reflexes control slower turns (Hengstenberg, 1993; Sherman and Dickinson, 2003; Srinivasan and Zhang, 2004).
 - To acknowledge the fact that only optical flow generated by translation contains information about the distance to objects (observation 2, section 2.3), flies and

honeybees show a flight behavior where periods of straight flights and rapid turns alternate (Collett and Land, 1975; Wagner, 1986; Schilstra and Hateren, 1999; Tammero and Dickinson, 2002a). During the straight flights only translational optical flow is perceived that can be used for relative distance estimation.

- Spontaneous turning maneuvers found in the flight behavior of flies (Heide, 1983) might help to overcome the problem that small objects in the line of flight do not generate a lot of optical flow (observation 4, section 2.3). By spontaneously changing flight direction objects are seen at a different angle and those objects that before the change of flight direction did not generate sufficient optical flow for a detection can be detected afterwards. The spontaneous turning maneuvers also make it more difficult for a predator to approach a prey by performing some motion camouflage where the predator uses the fact that objects moving with the observer do not cause optical flow.
- However, it seems that the ambiguity that the same optical flow can be generated either by the observer's self-motion or by moving objects in the observer's environment (observation 8, section 2.3) cannot be resolved by insects and is consequently used by experimentalist to study the visual system and behavior of those insects (David, 1982; Srinivasan et al., 1991; Krapp and Hengstenberg, 1996; Krapp et al., 1998; Fry et al., 2009). By stimulating insects with visual pattern similar to those they would perceive during self-motion the insects' flight behavior can be manipulated and studied.

What remains to be understood is how insect can provide such robust optical flow measures in the presence of low-contrast stimuli. Si et al. in 2003 characterized the effect of stimulus contrast and spatial frequency on the distance measure by honeybees. To their surprise they found that the bees' distance measures are "*relatively unaffected by variations in the contrast and spatial frequency of the visual patterns*" presented to the bees. A strong signal for the distance measure was "*generated even when the walls or the floor of the tunnel provided only low-contrast optical-flow cues*" (Si et al., 2003). This robustness of the optical flow sensor system of honeybees is a key property to achieve reliable and robust behavioral control. Information about how flies and bees achieve this robustness is limited.

However, providing such robust optic flow output signals that are independent from stimulus contrast and spatial frequency is also a key property of optical flow sensors to be attractive for MAV control. However, as described in chapter 6, this robustness is not supported by many aVLSI implementations.

Chapter 4

Optical flow based guidance of micro aerial vehicles

Abstract

This chapter presents three examples of optical flow based guided MAVs. The chapter's goal is to gain a deeper understanding of the existing flying platforms, their flight properties, controllers and which challenges these platforms present to optical flow sensors. Furthermore it defines the target platform for the sensors developed in this thesis. From the many existing optical flow controlled MAVs in the literature (Green et al., 2003, 2004; Ruffier and Franceschini, 2004, 2005, 2008; Chahl et al., 2004; Srinivasan et al., 2009, 2011) the 10-gram indoor airplane by Zufferey et al. (2007), the outdoor airplane equipped with commercial optical mouse sensors by Beyeler et al. (2009), and the quadcopter by Zingg et al. (2010) that uses a commercial camera with a fish-eye lens have been selected for presentation.

4.1 Wall-avoidance with 10-gram indoor airplane and single linear camera

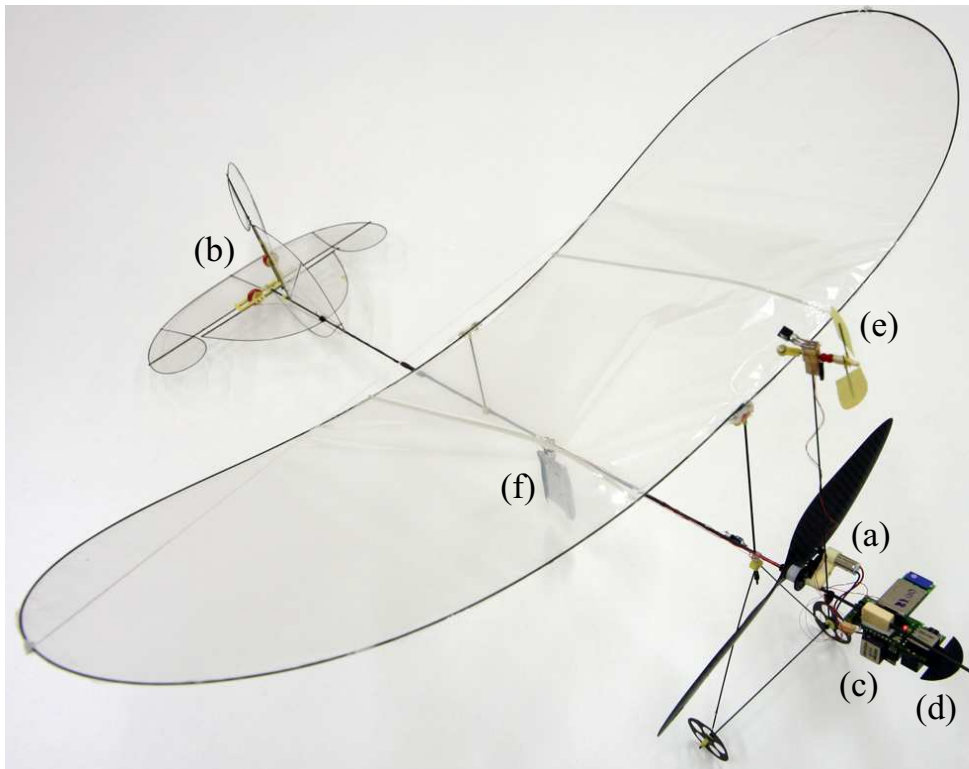
From 2001 until 2007 the Laboratory of Intelligent Systems, EPFL, published a series of papers on light-weight indoor airplanes (Nicoud and Zufferey, 2002; Zufferey and Floreano, 2005, 2006; Zufferey et al., 2006). So far, the most advanced prototype is the 10-gram indoor airplane (Zufferey et al., 2007) shown in figure 4.1a. This airplane is equipped with two cameras - one pointing downwards and one forward that is used for

autonomous wall-avoidance based on optical flow. The airplane is furthermore capable of automatic take-off and speed regulation.

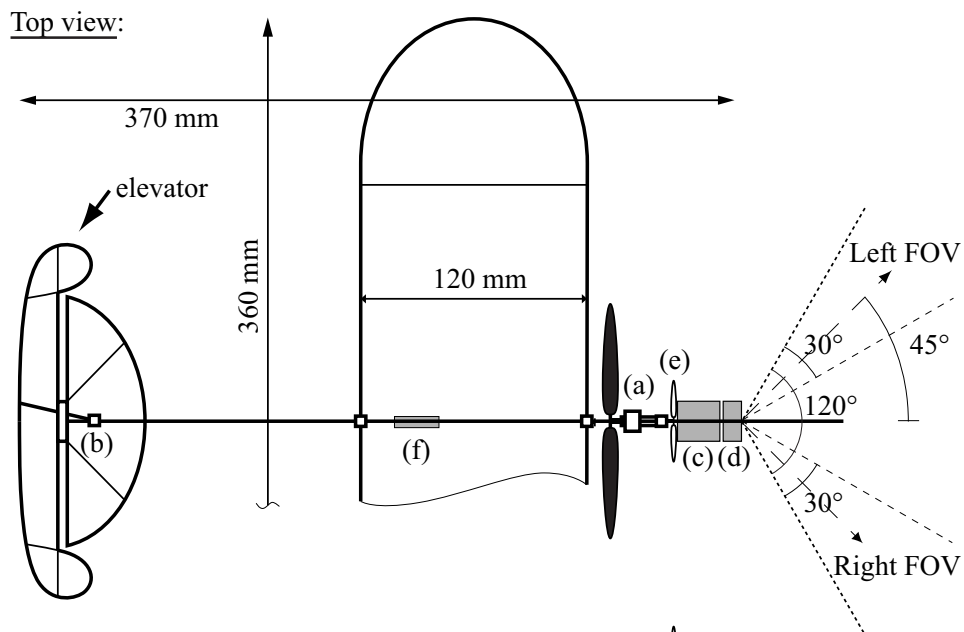
An airplane that should fly indoors in small rooms without the capability to hover like a helicopter has to be slow-flying - ideally flying at a speed of only a few meters per second. Such a slow-flying airplane needs to be light-weight to provide sufficient lift for take-off and altitude stabilization at such low speeds. So building and controlling an autonomous 10-gram airplane flying at around 1.5m/s like the one presented by Zufferey et al. (2007) is challenging.

To meet the weight requirements the airplane by Zufferey et al. (2007) is mainly made out of carbon fiber rods and thin Mylar plastic films. A picture of the airplane as well as a top view schematic is shown in figure 4.1. Despite its small weight, the airplane contains the necessary sensors and microcontroller for autonomous speed control and obstacle avoidance. A schematic of the implemented controller can be found in figure 4.2a. An on-board anemometer measures the airflow and is used to regulate the airplane's propeller velocity to keep the flight speed constant. Wall avoidance is implemented through the front camera module shown in figure 4.2c. This module contains a standard linear camera with 80 pixels from which as shown in figure 4.1b only the 20 pixels with a visual field of view between -30° and -60° (left side of airplane) and those 20 pixels between 30° and 60° (right side of airplane) are read by the microcontroller. The microcontroller which is running as well the flight controllers, computes two separate optical flow values OFR and OFL for the camera pixels pointing to the left and right using a linear version of the image interpolation algorithm by Srinivasan (1993, 1994). From the observations in section 2.3 we know that only the translational component of the total optical flow detected by the camera will contain information about wall distance. So Zufferey et al. (2007) subtracted the measurements of a rate gyroscope attached to the camera module as shown in figure 4.2c from the two optical flow values OFR and OFL. The difference of the remaining translational optical flow components TOFR and TOFL called OFDiv is then used to control the airplane's rudder. The implemented controller leads to the following behavior: If the translational optical flow detected on one side of the airplane which due to the constant speed of the airplane is inversely proportional to wall distance (see section 2.3) is above a threshold the airplane will steer towards the opposite direction and turn away from the wall.

I found this airplane by Zufferey et al. (2007) very impressive to work with. For a long time it seemed technically too difficult to allow such a low-weight flying platform to be equipped with sufficient sensors to perform automatic wall avoidance. Autonomous flight of this airplane has been demonstrated in a 7m x 6m room with random high-

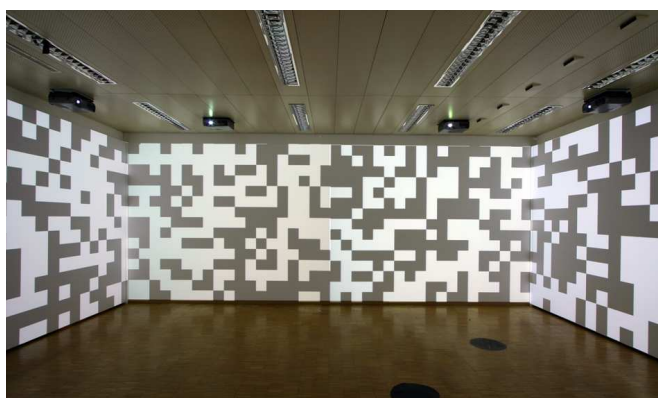
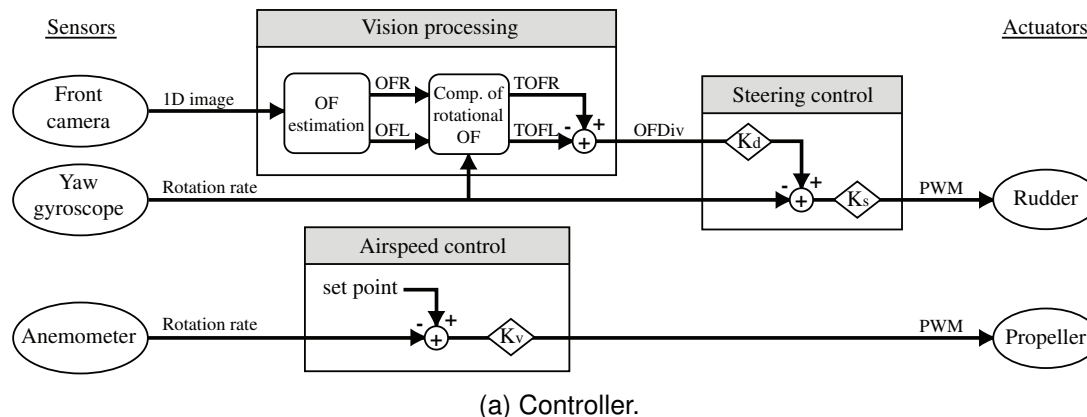


(a) Picture of airplane.

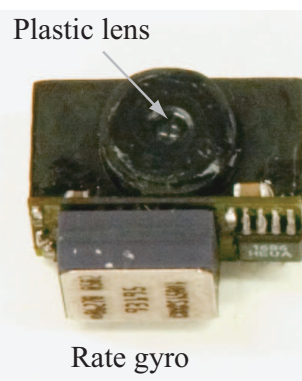


(b) Top view schematic.

Figure 4.1: (a) Picture of the airplane by Zufferey et al. (2007) and (b) a top view schematic. Figures (a) - (b) are adapted from Zufferey et al. (2007).



(b) Experimental room.



(c) Camera module.

Figure 4.2: (a) Controller implemented on the airplane for speed control and wall avoidance. (b) Experiments have been carried out in a 7m x 6m room equipped with 8 projectors that can project arbitrary pictures at the walls. (c) 0.9-gram camera module containing an off-the-shelf linear image sensor (TSL3301 from TAOS, Inc) in a custom plastic package and a rate gyroscope (ADXRS150 from Analog Devices Inc) for compensation of rotational optical flow. Figures (a) - (c) are adapted from Zufferey et al. (2007).

contrast black and white checkerboard patterns being projected onto the walls (see figure 4.2b). Maneuvering in such a small space is challenging with an airplane that cannot hover like a helicopter. The airplane successfully autonomously turns away from walls that are too close but still requires a human-in-the-loop for altitude control. This is because airplanes typically tend to roll when turning - by this losing altitude during the maneuver that needs to be regained after each turn. With its limited sensing capabilities, the airplane by Zufferey et al. (2007) is not able to control its altitude autonomously and thus requires a human for altitude correction.

A drawback of the light-weight construction is that the airplane can only carry little

payload and is relatively fragile - making it difficult to use in scientific experiments. During experiments I found that the airplane breaks easily when crashing into walls - requiring repair -, needs quite some training to be controlled by a human and can only fly when there is very little air turbulences like wind.

A further limitation is the used camera module. Due to its limited capabilities the airplane performs well only with high-contrast stimuli and good background illumination. So I believe that the airplane could take great benefit from replacing the linear camera a custom aVLSI optical flow sensor that works with low-contrast stimuli, under a variety of background illuminations and that frees the microcontroller from the computationally expensive task of reading local pixel values from the camera and computing the optical flow values.

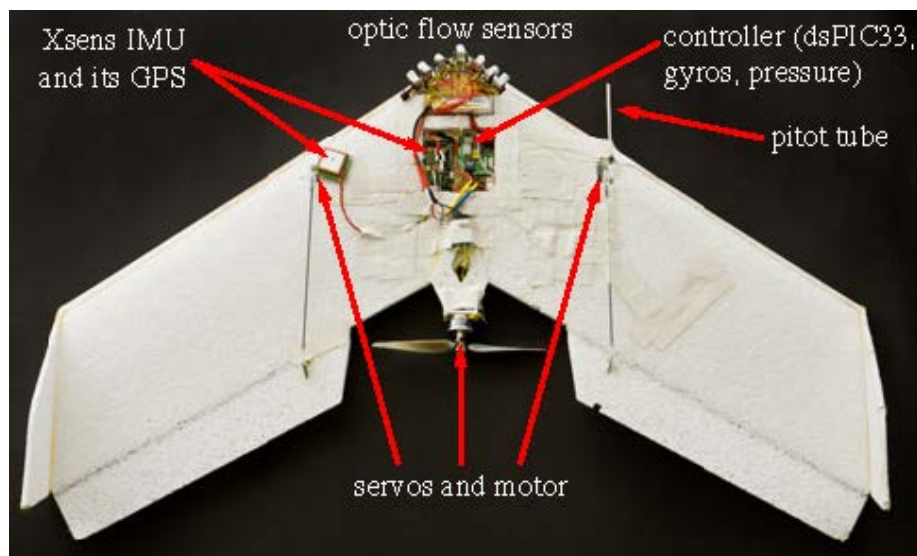
Nevertheless, the presented level of autonomy and the successful demonstration of wall avoidance with such minimalistic control and light-weight sensors make the airplane a great success.

4.2 Obstacle avoidance with outdoor airplane and 7 optical mouse sensors

In 2009 our colleagues from the Laboratory of Intelligent Systems at EPFL presented a fully autonomous outdoor airplane that only relied on gyroscopic information and optical flow extracted by seven off-the-shelf optical mouse sensors (Beyeler et al., 2009).

The airplane presented by Beyeler et al. (2009) is a modified flying Swift wing shown in figure 4.3a (Leven et al., 2009). The flyer has a wing span of 80cm and a total weight of 407g. What makes the flyer interesting for flight control is that it is neutrally stable in roll and pitch. After a disturbance around the roll or pitch axis the flyer will not continue rotating but continue translating in the direction reached due to the disturbance.

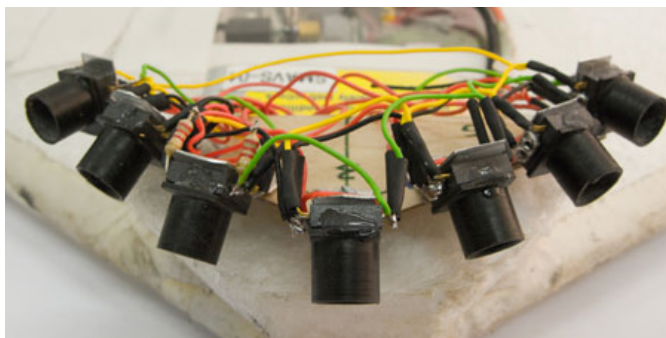
To keep the airplane's velocity close to 14m/s, Beyeler et al. used a differential pressure sensor (MPXV5004DP from Freescale) for speed control. Furthermore, three rate gyroscopes (ADXRS610 from Analog Devices Inc.) were mounted for derotation of the optical flow estimated by the optical mouse sensors to extract only the translational distant-depended component of the total optical flow. All computation was done on-board with a 40MIPS dsPIC microcontroller from Microchip. For optical flow estimation Beyeler et al. added an array of seven optical mouse sensors. See figure 4.3b for a picture of a single 0.8-gram optical flow sensor based on the ADNS-5050 off-the-shelf



(a) Airplane.



(b) Optical mouse sensor.



(c) Sensor array.

Figure 4.3: (a) Modified Swift wing flyer with optical flow sensor array. (b) Single 0.8-gram optical flow sensor consisting of off-the-shelf optical mouse sensor (ADNS-5050 from Avago) and custom optics based on the Philips CAX 100 collimator lens ($f = 10\text{mm}$). (c) Optical flow sensor array composed of seven sensors from (b) mounted on wing flyer. Figures (a) - (c) are adapted from Beyeler et al. (2009).

optical mouse sensor from Avago and the Philips CAX 100 collimator lens and figure 4.3c for a picture of the sensor array.

The optical mouse sensors were arranged as shown in figure 4.4 so that two weighted sums of the individual optical flow signals could be used for controlling the pitch and roll of the airplane. The directions in which the optical mouse sensors were pointing were chosen to roughly mimic the wide field of view and function of the vertical cells in flies that are believed to estimate self-motion (see section 3.4 and Krapp et al. (1998)). For

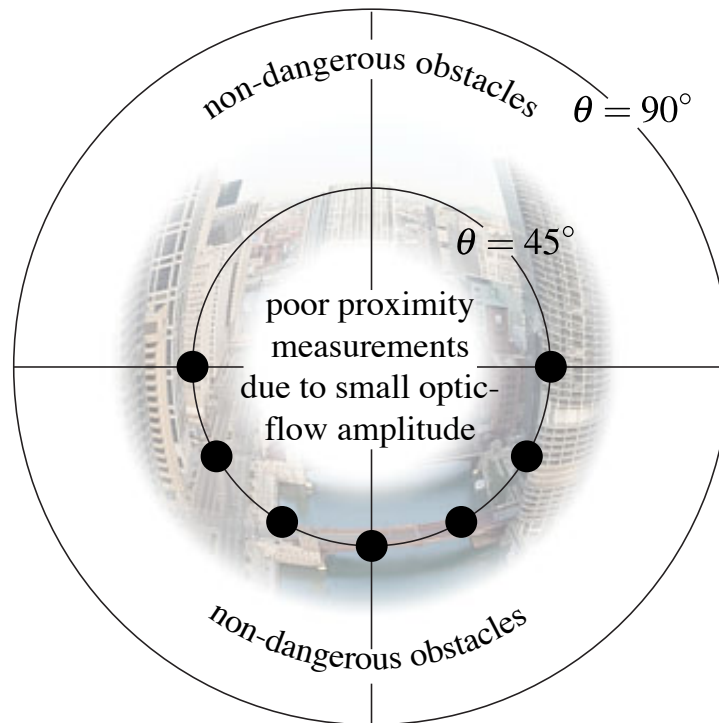


Figure 4.4: Relevance of regions of the field of view for optical-flow based control of the airplane. During normal flight the airplane is expected to generate translational optical flow fields forming expansion patterns. Optical-flow estimates in directions with an angle $\hat{\theta}$ relative to the flight direction (center of the field of view) much smaller than 45° are thus expected to be of small amplitude. Obstacles in directions much greater than 45° mean no threat to the airplane and are thus not relevant for the flight controller. So Beyeler et al. placed seven optical mouse sensors pointing in the directions depicted as black circles with $\hat{\theta} = 45^\circ$. The figure is adapted from Beyeler et al. (2009).

choosing the number of optical mouse sensors to be used Beyeler et al. (2009) had to do a trade-off between the accuracy of the optical flow estimation and the payload created by the additional sensors. As few sensors as possible were desired. Beyeler et al. decided to place seven optical mouse sensors pointing in the directions depicted as black circles in figure 4.4. Optical mouse sensors point in the direction with a polar angle $\hat{\theta} = 45^\circ$ relative to the flight direction of the airplane which corresponds to the center of the figure 4.4. As we know from the expansion patterns shown in figures 2.6a and 2.6b in section 2.3, we can expect the optical flow close to the center of the field of view to be of small amplitude. This is why no optical mouse sensor pointed in this direction. Since the aircraft typically only creates thrust it is very unlikely to expect anything except expansion patterns after the derotation of the optical flow if only the

aircraft is allowed to move but not the objects in its environment. Obstacles detected at large angles $\hat{\Theta}$ relative to the direction of flight are considered to be irrelevant for control since they would be no threat to the airplane. So Beyeler et al. found the angle of $\hat{\Theta} = 45^\circ$ to be a good trade-off for placing the optical mouse sensors. Sensors were only placed to look towards the ground and not towards the ceiling since the airplane is flying outside and the sky is considered to generate only translational optical flow close to zero.

When I had the chance to manually steer the wing flyer used by Beyeler et al. (2009) I found it very stable and impressive to fly. The fact that it is neutrally stable and very robust to external perturbations makes it very user friendly. After changing the airplane's pitch or roll angle the airplane stably proceeds flying on its new course. The OptiPilot controller presented by Beyeler et al. (2009) successfully restores the pitch and roll angles as well as the flight altitude after a manually caused perturbation to a given set-point. Furthermore, the airplane was demonstrated to autonomously avoid sufficiently big obstacle. I thus believe that this airplane is currently the state-of-the art in its field.

Beyeler et al. furthermore extensively tested the OptiPilot in simulation and generated only 16 collisions in 7 hours flight. Fully autonomous long term flight experiments like those in simulation have not been presented yet with the real platform. This might be since it is technically challenging but also since the optical mouse sensors could easily fail outside when confronted by non-ideal textures and lighting conditions which are not tested yet. Since optical mouse sensors are typically optimized to deal with low contrast under constant lighting conditions, they seem to be well suited to fly over low contrast areas in daylight. From my experience the optical mouse sensors e.g. perform well when looking at grass or trees.

4.3 Passing corridors with quadcopter and single camera with fish-eye lens

Zingg et al. (2010) presented an optical-flow based approach for safely maneuvering a quadcopter through a corridor. The demonstration was not fully autonomous but showed a different approach than the ones presented before in this chapter using a camera with a wide field of view.

Zingg et al. (2010) modified the Hummingbird quadcopter by Ascending Technologies (figure 4.5a) by adding a downward pointing μ Eye camera from IDS as shown in figure 4.5b at the bottom of the platform. Due to a fisheye lens the camera with a spatial

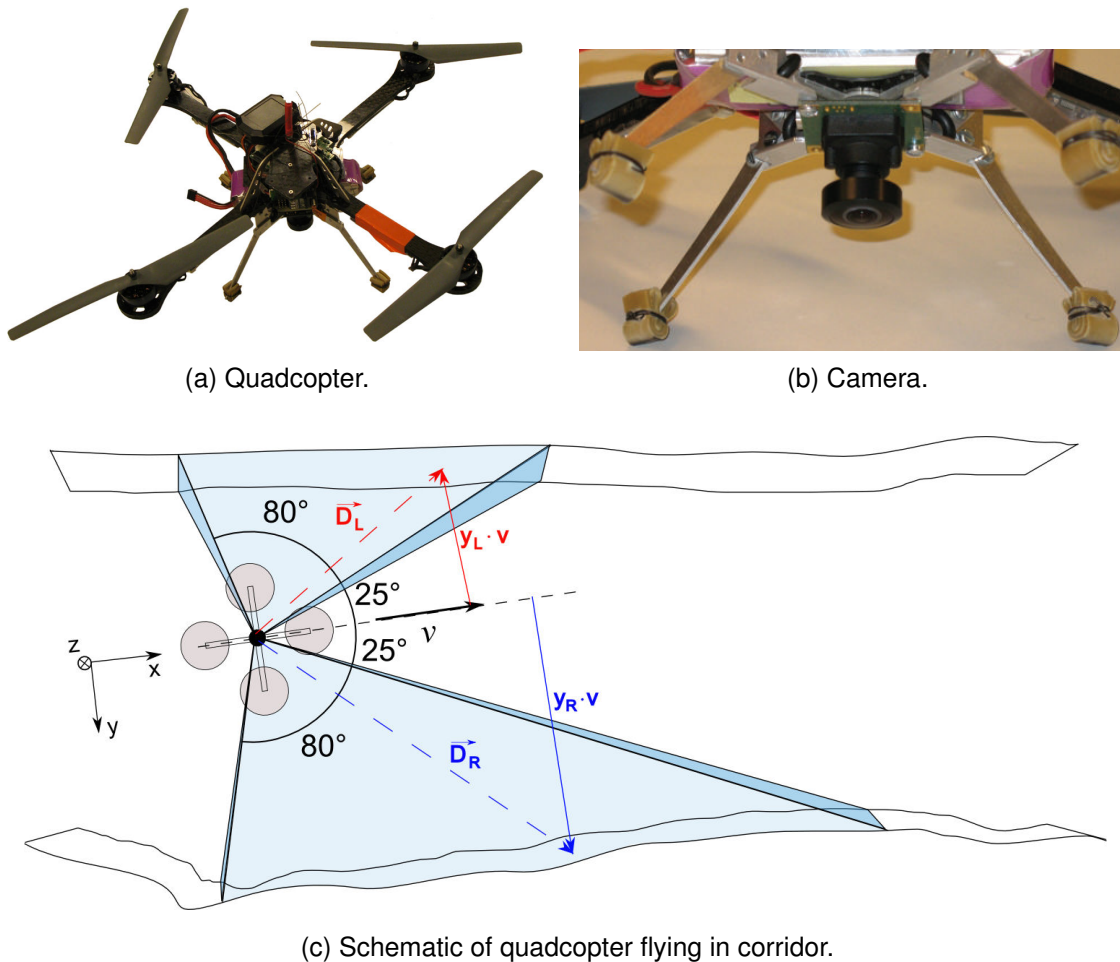


Figure 4.5: (a) Hummingbird quadcopter by Ascending Technologies used during the experiments. (b) μ Eye camera from IDS mounted on quadcopter. (c) Schematic of the quadcopter flying through a tunnel. For optical flow computation Zingg et al. (2010) only used two 80°-patches from the overall camera's field of view pointing to the left and the right of the quadcopter. Figures (a) - (c) are adapted from Zingg et al. (2010).

resolution of 752 x 480 pixels and a frame rate of 87 frames per second had a field of view of 190°. For extraction of optical flow Zingg et al. (2010) used two 80°-patches from the total field of view that were used to estimate the wall distance of the corridor on the left and right side of the quadcopter. Due to the high computational cost of the optical flow calculation from the camera images, images from the patches were sent to an external computer with a 2-GHz CPU via a wireless ZigBee communication link.

Zingg et al. (2010) computed optical flow from consecutive camera frames at 20 Hz using Shi and Tomasi's corner finder (Shi and Tomasi, 1994) for selecting pixels in frame 1 and the Lucas-Kanade method (Lukas and Kanade, 1981; Bouguet, 2001) for finding

the corresponding pixels in frame 2. On average 250 optical flow component have been calculated for the left and right 80°-patches and for each pair of frames. An on-board Inertial Measurement Unit (IMU) was used to remove the rotational component of the computed total optical flow and to form two relative depth maps from the translational optical flow extracted from the left and right side of the quadcopter. For conversion of the deformed images due to the fish-eye lens Zingg et al. used a self-developed toolbox for calibrating omni-directional cameras (Scaramuzza et al., 2006). Furthermore, the calculated optical flow components are inspected for outliers to remove wrong matches in the Lucas-Kanade method.

Zingg et al. (2010) did not demonstrate autonomous optical-flow based flight control but instead showed data where they manually made the quadcopter fly through a corridor and compared the estimated to the actual externally measured wall distances. The computed error of the estimated distance was very noisy and could have a maximum deviation from the ground truth of about a quarter of the corridor width. The fact that Zingg et al. (2010) used a fisheye lens to generate an optical flow sensor with a wide field of view corresponding to a virtual array of on average 500 optical flow sensors made the publication interesting to present and to compare with.

4.4 Conclusion

An overview of properties of the presented optical-flow controlled MAVs can be found in table 4.1. In summary one can state that the so far most successful implementations of optical-flow based control have been presented on airplanes but attempts and demonstrations of optical-flow based altitude control have been shown as well on helicopters (Netter and Francheschini, 2002; Ruffier and Franceschini, 2004, 2005, 2008). More recently quadcopters have as well become popular research demonstration tools (Zingg et al., 2010). The success of airplanes is to some extent due to their well understood flight properties with limited amount of degrees of freedom. It is for example easier to control an already passively stable airplane where only thrust, roll and pitch need to be controlled than a passively unstable quadcopter that typically drifts and has six degrees of freedom (thrust, slip, lift, roll, pitch, yaw). Nevertheless when the stability problem is solved helicopters and quadcopters are interesting research platforms that can fly both very slowly and fast indoors and outdoors and that are capable of carrying lots of payload. An example for a commercially available quadcopter with inertia sensors, ultra-sonic distance sensors and image processing is the AR.Drone from Parrot. Due to multi-sensor fusion the AR.Drone is actively stabilized. Especially due to its image

Table 4.1: Comparison of properties of optical-flow controlled MAVs presented in this section.

	Indoor airplane by Zufferey et al. (2007)	Outdoor airplane by Beyeler et al. (2009)	Quadcopter by Zingg et al. (2010)
Optical flow sensor	linear camera	array of 7 optical mouse sensors	camera with fisheye lens
Total size	37 x 36cm	80cm wingspan	Ø53cm
Total weight	10.3g	407g	unknown
Payload	<2g	>50g	200g
Sensor weight	0.8g	7 x 0.8g = 5.6g	>20g
Flight speed	1.5m/s	14m/s	unknown for experiment (50km/h max)
Optical flow algorithm	Image Interpolation Algorithm	unknown/integrated in optical mouse sensor	Lucas-Kanade method
Processor	32MHz/10MIPS on-board	80MHz/40MIPS on-board	2GHz dual core off-board
Wireless interface	Bluetooth	2.4 GHz Digi XBee-PRO radio-link	ZigBee

processing that as well involves optical flow estimation the AR.Drone is relatively stable and especially works well when its bottom camera sees high contrast patterns and when it is operated on flat terrain. However, when flying over uneven terrain in terms of stability the wing flyer used by Beyeler et al. (2009) is still outperforming.

For the measurement of optical flow **standard cameras are still the most dominant** the presented examples have been selected because they demonstrate three different techniques to gather the necessary global optical flow measurements for successful control of the MAV. The most minimalistic approach so far is the estimation of two optical flow measures for obstacle avoidance on the left and right side of the airplane with a single linear camera presented by Zufferey et al. (2007). However, this approach does not allow for autonomous altitude control. More typical approaches involve a standard camera in combination with a wide-angle fisheye lens like presented by Zingg et al. (2010). Due to the typically more heavy optics of fisheye lenses this approach cannot be taken by very light-weight MAVs but in terms of space and weight there is a certain

trade-off between the use of a wide-angle lens and several cameras with small-angle optics. In 2007 Soccol et al. presented another approach to measure global optical flow with a standard camera and a specially shaped mirror that projects the perceived images onto the camera in a way that is advantageous for estimation of optical flow. The mirror approach is the most heaviest presented in this section and so far it was shown that optical flow could be measured on an outside airplane but no autonomous control has been presented (Srinivasan et al., 2009, 2011). The most autonomous optical flow controlled MAV presented so far is the wing flyer by Beyeler et al. (2009) since it performs both obstacle avoidance and altitude control.

This thesis is part of a project founded by the Swiss National Science Foundation (grant number: 200021-105545/1) in collaboration with the Laboratory of Intelligent Systems. As part of this project the indoor airplane by Zufferey et al. (2007) had been selected as the reference target flying platform for the development of the optical flow sensors presented in this thesis since it was the most autonomous optical flow controlled MAV at that time. The presented optical flow sensor modules (chapters 9 and 7) have been optimized for operation on this airplane.

Zufferey et al. (2007) have successfully demonstrated that for such airplanes **linear cameras or optical flow sensors can be sufficient**. Two-dimensional sensors are not necessary. Also the airplane by Beyeler et al. (2009) could be operated with linear sensors. As presented in chapters 9 and 7 such linear sensor arrangements are of advantage for the design of the sensors' layout since one can place a one-dimensional array of narrowly spaced photodiodes supporting good spatial resolution of the sensor with the circuits for optical flow computation on the side.

Non-negligible remains for optical-flow based MAV control the **high computational cost for optical flow computation**. This cost forces MAV designers still to preform the computation of optical flow off-board on an external PC when using cameras of higher resolution as for example presented by Zingg et al. (2010) or when using more demanding flight controllers. The minimalistic approach by Zufferey et al. (2007) stays an exception in the literature. The use of several dedicated optical flow sensors like the optical mouse sensors presented by Beyeler et al. (2009) has the advantage that these optical mouse sensors already take care of the full image processing directly outputting the optical flow estimates. However, using fixed optical mouse sensors can be very limiting since customization is almost impossible. This is why this thesis is targeting at the design of custom optical flow sensors that partially or fully handle the image processing, thus reducing the need for additional computational resources - allowing for the integration of the entire control and vision system on the MAV.

An advantage of the specialized optical flow sensor modules is their design aiming at applications in **low contrast** environments with a **large deviation of background light intensity**. This combination is not typical for standard camera or standard optical mouse sensors but is one of the limiting factors for the application of MAVs in a variety of challenging natural and indoor environments which so far has not been demonstrated by autonomously flying MAVs. In fact artificial high contrast pattern like those presented by Zufferey et al. (2007) or simple black-and-white stripes are the most dominant stimuli used in the literature so far. The airplane by Beyeler et al. (2009) flying outside while using natural stimuli is still an exception.

Chapter 5

A short introduction to analog VLSI technology

Abstract

This chapter gives a short introduction to analog Very Large Scale Integrated (aVLSI) technology to allow a non-chip designer to understand the circuits described in the following chapters and to give some insights into the challenges of aVLSI chip design in general. aVLSI technology allows the integration of several million devices on a single chip within a few square millimeters and thus for the design of small, light weight, highly integrated sensors and computing devices. The term "analog" in aVLSI refers to the fact that one is not using circuits to only form digital systems but that one uses the analog transfer function of the basic devices in this technology.

This chapter is not intended to be a complete handbook for chip design. For more detailed information about aVLSI chip design, see for example, the book "Analog VLSI: Circuits and Principles" by Liu et al. (2002).

Section 5.1 gives a short introduction into the basic elements used for chip design: transistors, capacitors and resistors. Section 5.2 discusses techniques to deal with device mismatch caused by imperfections in the fabrication process of aVLSI chips while section 5.3 discusses different types of circuit noise. Section 5.4 concludes this chapter.

Addressed questions

This chapter addresses the following questions:

1. What are the basic elements used in aVLSI technology and what are their properties? (section 5.1)
2. What is circuit mismatch and how can it be reduced or compensated? (section 5.2)
3. What are the types of noise in circuits? (section 5.3)

5.1 The basic elements for chip design: transistors, capacitors, resistors

This section introduces the three major devices in aVLSI circuits: transistors, resistors and capacitors. The understanding of their basic properties is crucial for the design of aVLSI sensor like optic flow sensors.

5.1.1 CMOS field effect transistors

"Transistor" is a generic term describing a solid-state device with three or more terminals. In the circuits described in this thesis, I use so-called Metal Oxide Semiconductor Field Effect Transistors (MOSFET). The MOSFETs' layout structure and schematic symbols are shown in figure 5.1. MOSFETs are devices with four terminals: source, drain, gate and bulk.

The term "Semiconductor" in MOSFET stands for the fact that MOSFETs electrically can both behave like conductors and insulators. The term "Field-Effect Transistor" (FET) refers to the MOSFETs' property that the electric field between the MOSFETs' gate (G) and source (S) terminal modulates the amount of current that is flowing from the source (S) terminal through the MOSFETs' channel to the drain (D) terminal. For a small gate-to-source voltage (V_{GS}) only a small current (I_{DS}) in the range of femto to microamperes flows between source and drain. For high V_{GS} , a channel is formed between source and drain and I_{DS} can be in the range of several hundred microamperes.

The term "Metal Oxide" in MOSFET describes the thin insulation layer composed of a metal oxide that insulates the MOSFET's gate from its channel. In contrast to bipolar

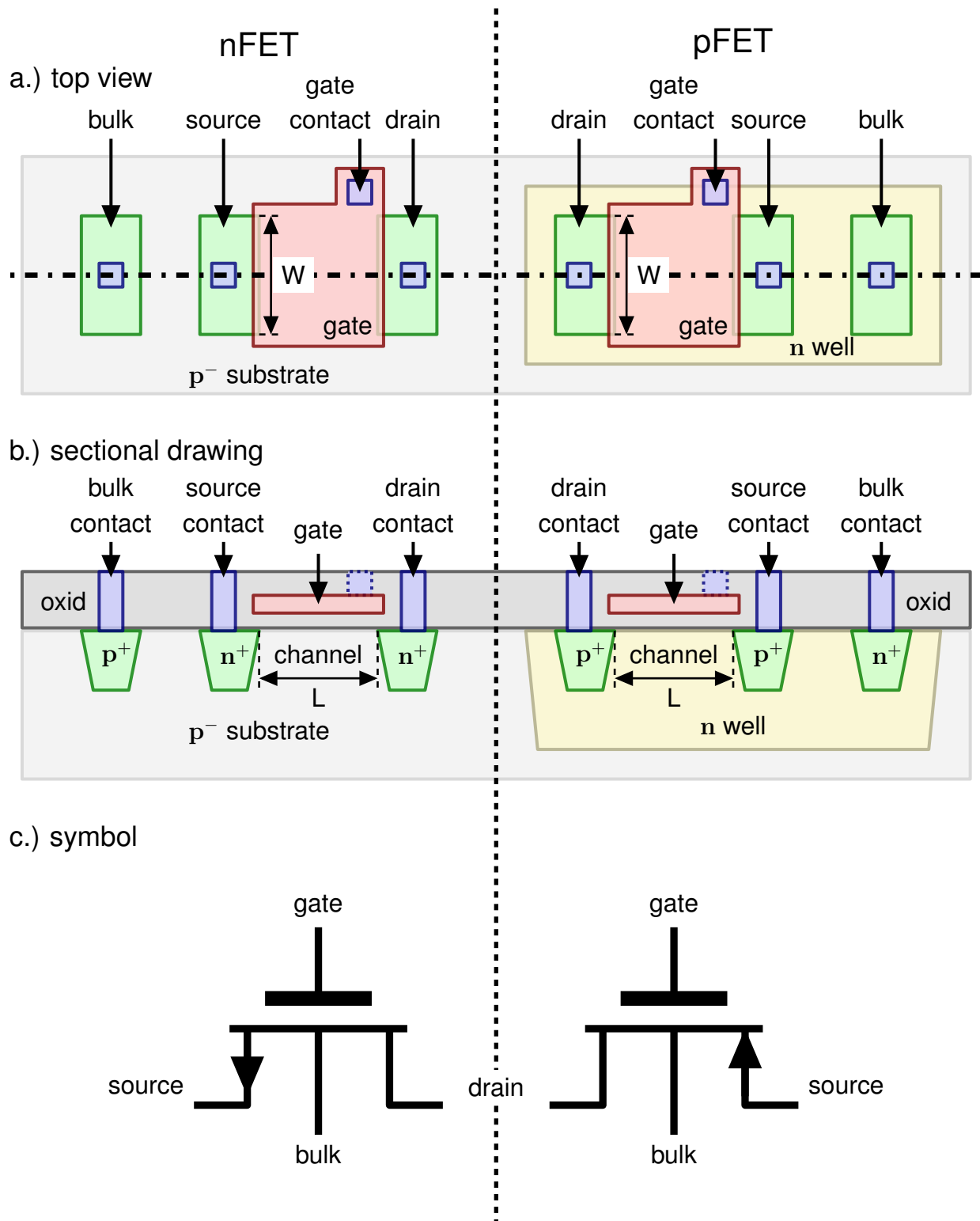


Figure 5.1: Layout, structure and symbol of nFET and pFET.

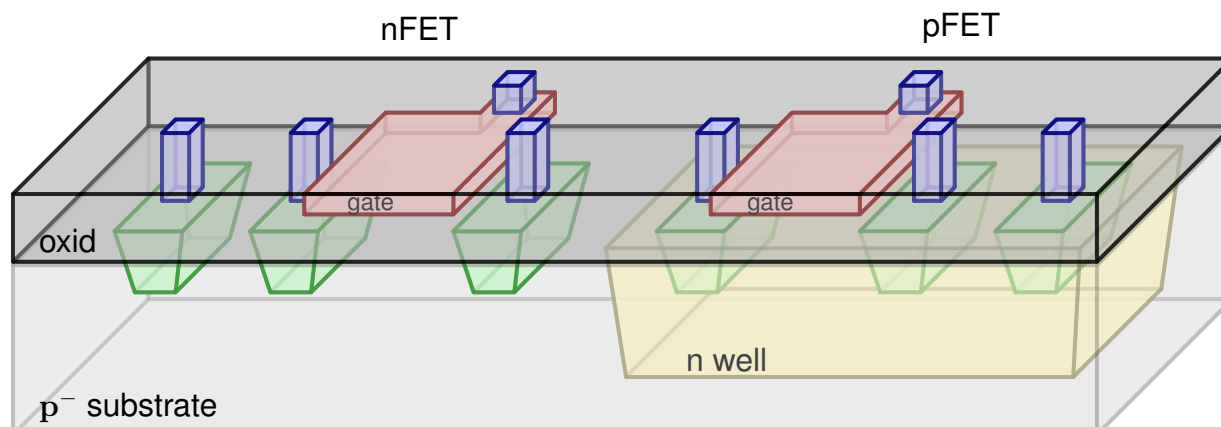


Figure 5.2: 3D schematic of nFET and pFET layout integrated on same substrate in CMOS technology.

transistors in MOSFETs, no current flows into the MOSFET's gate. This is one of the reasons why MOSFETs consume much less power than bipolar transistors making MOSFETs suitable for very large scale integrated systems. The term "Metal Oxide" is nowadays not fully correct anymore since in more modern fabrication processes like the ones being used for chips described in this thesis the metal oxide insulator is being replaced by an insulator composed of silicon dioxide (SiO_2). The term "Metal Oxide" refers to the time of the first MOSFETs where the insulator was still fabricated from aluminum.

FETs were first conceived by Lilienfeld who filed a patent in October 1926 which was granted in 1930 (Lilienfeld, 1930). However, it took until 1959 when Atalla and Kahng built the first working insulated-gate field-effect transistor at Bell Labs. Kahng filed a patent describing their invention in 1960 which was issued in August 1963 (Kahng, 1963). Several implementations of MOS transistors followed like the MOS-controlled tetrode by Sah (1961) and a 16-transistor integrated device by Hofstein and Heiman (1963).

A reason why mass production of VLSI chips became feasible was the development of the Complementary Metal Oxide Semiconductor (CMOS) technology (Wanlass and Sah, 1963; Wanlass, 1967) that allowed the design and fabrication of digital circuits that consume power close to zero in standby mode. The term "Complementary" in CMOS refers to the fact that the technology allows the fabrication of two different kinds of complementary transistors on the same chip: n-channel FETs (nFET) and p-channel FETs (pFET). In nFETs the majority carriers are negatively charged electrons that flow

from most negative terminal (source of nFET) to most positive terminal (drain of nFET) through the nFET's channel. The majority carriers of pFETs are positively charged holes that flow from most positive terminal (source of pFET) to most negative terminal (drain of pFET) through the pFET's channel. Figure 5.1 shows layout, structure and symbols for both nFET and pFET. Figure 5.2 shows a 3-dimensional view of an nFET and pFET integrated on the same substrate. The graphics visualizing the transistors' structures show that the transistors' drain and source terminals are physically equal. Just the potential that the terminals are connected to during the operation of the transistor defines which side of the transistors' channel is called the source and which is referred to as the transistors' drain. The integration of both nFETs and pFETs onto the same VLSI chip became possible by the fabrication step of wells that separate the different FETs.

While in digital CMOS designs the MOS transistors are used as binary switches, analog circuit designers make use of the analog properties of the MOS transistors. This is the reason for the variety of analog functions that can be implemented with these devices.

Figure 5.3 shows simulation results of an nFET's drain-to-source current I_{DS} for changing gate-to-source voltages V_{GS} while the drain-to-source voltage V_{DS} of the transistor is kept constant. Figure 5.4 shows how I_{DS} varies with changing V_{DS} for several constant V_{GS} values. The simulation was performed with a SPICE (Simulation Program With Integrated Circuit Emphasis) simulator. The nFET's simulation parameters were chosen according to the AMS $0.35\mu m$ process that was used for the adaptive vision sensor presented in chapter 9.

While the SPICE simulation software uses a more complex and accurate model of the transistors' behavior, we first consider a more simple first order model for the analytical description of the circuits presented in this thesis. The derived circuit behavior from the simple model is then verified through the SPICE simulations.

To first order, four different operating regimes of transistor can be distinguished. We first distinguish between the subthreshold regime where the transistor current, I_{DS} , is due to diffusion of carriers from source to drain. This current is exponentially dependent on V_{GS} . In the above threshold regime, I_{DS} is due to drift of carriers. This current is quadratic in V_{GS} . For both subthreshold and above threshold regimes, the operating regime can be subdivided again into a triode / linear region where I_{DS} is dependent on both V_{GS} and V_{DS} , and a saturation region where I_{DS} is independent of V_{DS} and dependent only on V_{GS} .

Equations describing the first-order behavior of nFETs and pFETs in the four regimes are shown in tables 5.1 and 5.2, respectively.

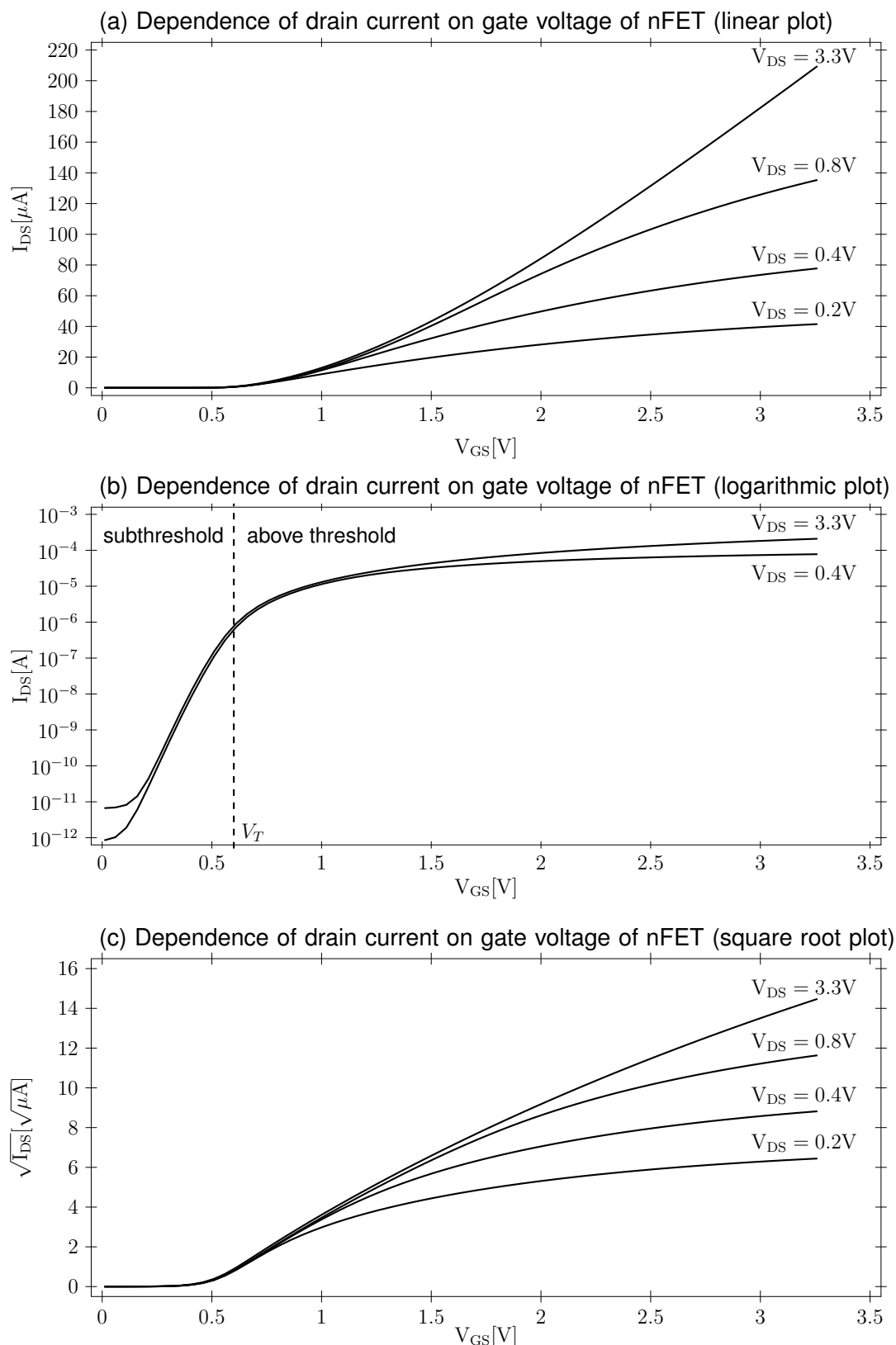


Figure 5.3: Characteristic curve of an nFET for changing gate voltage V_{GS} and different drain voltages V_{DS} . The transistors width and length are $0.6 \mu m$. Data taken from SPICE simulations with Tanner Tools.

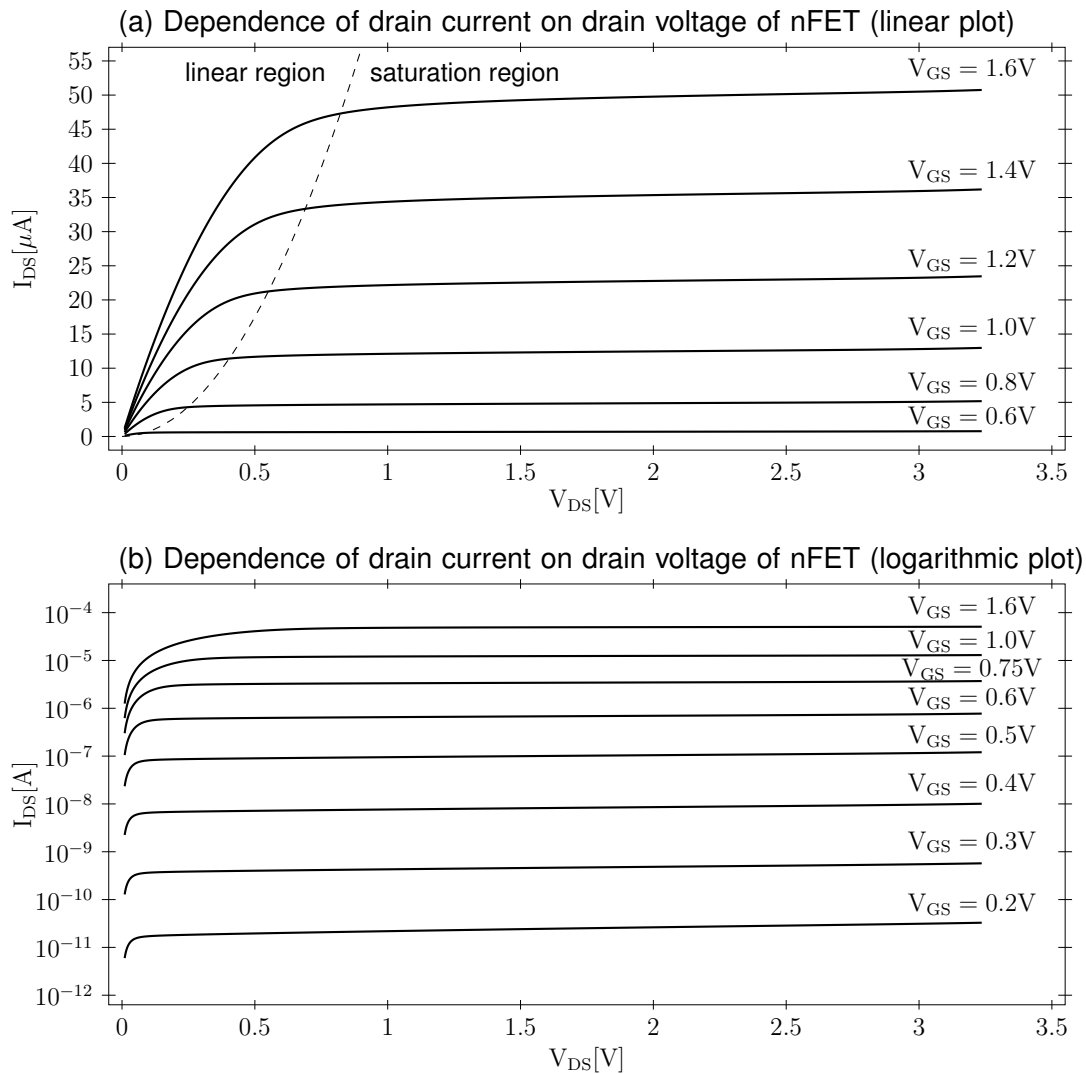


Figure 5.4: Characteristic curve of an nFET for changing drain voltage V_{DS} and different gate voltages V_{GS} . The transistors width and length are $0.6 \mu\text{m}$. Data taken from spice simulations with Tanner Tools.

Equations (5.1) and (5.3) describe the nFET's transfer function in the triode / linear and saturation region of the subthreshold regime, respectively, where

I_0 is the off-current and can be extracted from the place where the interpolated exponential intersects the current axis in the I_{DS} - V_{GS} plot,

κ is a parameter describing the efficiency of the gate voltage V_G in driving the channel and

U_T is the thermal voltage.

The subthreshold regime is the linear part of the transfer function in the logarithmic plot shown in figure 5.3b.

Table 5.1: Regimes of transistor operation: nFET

	Subthreshold / Weak inversion	Above threshold / Strong inversion
Triode/ Linear region	$I_{DS} = I_0 e^{\frac{\kappa V_G - V_S}{U_T}} \left(1 - e^{-\frac{V_{DS}}{U_T}} \right) \quad (5.1)$	$I_{DS} = \beta (V_{GS} - V_T) V_{DS} \quad (5.2)$
Saturation region	$I_{DS} = I_0 e^{\frac{\kappa V_G - V_S}{U_T}} \quad (5.3)$	$I_{DS} = \frac{\beta}{2} (V_{GS} - V_T)^2 \quad (5.4)$

Table 5.2: Regimes of transistor operation: pFET

	Subthreshold / Weak inversion	Above threshold / Strong inversion
Triode/ Linear region	$I_{DS} = I_0 e^{\frac{-\kappa V_G + V_S}{U_T}} \left(1 - e^{+\frac{V_{DS}}{U_T}} \right) \quad (5.5)$	$I_{DS} = \beta (V_{GS} - V_T) V_{DS} \quad (5.6)$
Saturation region	$I_{DS} = I_0 e^{\frac{-\kappa V_G + V_S}{U_T}} \quad (5.7)$	$I_{DS} = \frac{\beta}{2} (V_{GS} - V_T)^2 \quad (5.8)$

The nFET's transfer function in the triode / linear and saturation regions in the above threshold regime can be described by the equations (5.2) and (5.4), respectively, where V_T is the threshold voltage, and β is a parameter that relates the applied voltages to the drain-to-source current and that can be calculated from the transistor's width W , length L , oxide capacitance C_{OX} and the charge mobility μ according to

$$\beta = \mu C_{OX} \frac{W}{L} \quad (5.9)$$

The current is approximately a linear function of the gate and drain voltages in the triode region as shown in figure 5.3a. In the saturation region, the current follows a square law as shown in the linear part of the plot in figure 5.3c.

Although the equations for the nFET and pFET in the tables 5.1 and 5.2 are similar, the parameters of the nFET and pFET are typically different for a fabrication process. The mobility of the major carriers of pFETs is, for example, typically smaller than that of nFETs.

Figure 5.3b shows the separation between subthreshold and above threshold operation. As can be seen in the figure for normal transistors, there is a smooth transition between the two regimes of operation that is not reproduced by the first order model that assumes an abrupt change at the threshold voltage. This threshold voltage can be defined as the V_{GS} value where the measured I_{DS} is half of the current I_{DS} extrapolated from the exponential description in equation (5.3) (figure 5.3b). A model that includes the transition between the two regimes is the EKV model (cite Enz, Vittoz's paper).

The separation of the linear and saturation regions in the above threshold regime can be seen in figure 5.4a. Note that the boundary between both regions is not only dependent on V_{DS} but also on V_{GS} . One can roughly say that the transistor is operated in the linear region when $V_{DS} < V_{GS} - V_T$.

This variety of operation regimes of MOSFETs is the reason why analog circuit designers are able to build a variety of circuits like analog multipliers, adders, subtractors, linear and nonlinear filters, synapse circuits and integrate-and-fire neurons.

5.1.2 Capacitors

Capacitors are analog storage devices for charge. They can be physically implemented by two conducting materials that are separated by an insulator. The fabrication processes used for the aVLSI chips presented in this thesis allow for the implementation of two types of capacitors: poly1-to-poly2 and MOS capacitors. Poly1-to-poly2 capacitors are formed out of two polysilicon plates separated by oxide while MOS capacitors use the insulation layer of the gate oxide. The two plates consist of the gate of the MOSFET and the shorted source and drain of the transistor.

In general the capacitance C of any capacitor can be described as

$$C = \frac{Q}{V} \quad (5.10)$$

where Q is the amount of charge that is stored on the capacitor when a voltage difference V is applied to its terminals. From (5.10) one can derive an equation describing the amount of current I flowing into or out of a capacitor as a function of V .

$$I(t) = C \cdot \frac{dV(t)}{dt} \quad (5.11)$$

The capacitance C of a capacitor composed of two plane plates can be calculated as

$$C = \epsilon \cdot \frac{A}{D} \quad (5.12)$$

where ϵ is the permittivity of the insulator, A is the area and D is the distance between the two plates.

Table 5.3: Capacitance of poly1-to-poly2 and MOS capacitors for AMS $0.35\mu\text{m}$ process.

	Poly1-to-poly2 capacitor	MOS capacitor
Capacitance [fF/ $(\mu\text{m})^2$]	0.86	4.54

Table 5.3 shows the approximate capacitance per unit area values for poly1-to-poly2 and MOS capacitors in the AMS $0.35\mu\text{m}$ process that is used for the fabrication of the sensor presented in chapter 9. The table shows that with the same area one can achieve more than 5 times bigger capacitance by implementing the capacitor as a MOS-FET instead of a poly1-to-poly2 capacitor. However, capacitors are better matched by at least 1-2 decades over MOS capacitors (Minch et al., 1996). So the choice of which technique is used to implement a capacitor is a trade-off between layout area and matching requirements.

5.1.3 Resistors

Well-controlled resistors with high resistance are difficult to implement. They usually occupy a large chip area and have high parasitic capacitance which limits their application in high-frequency circuits. The AMS $0.35\mu\text{m}$ process supports the implementation

of high resistance polysilicon structures with $1.2\text{k}\Omega/\text{Area}$. This structure is used in the programmable bias-current generator circuit described in chapter 9.

There are various techniques to implement tunable resistors in the order of $\text{M}\Omega$ or $\text{G}\Omega$ through the use of MOS transistor circuits that are for e.g. operated in the linear region of the above threshold regime or translinear current mode circuits (Banu and Tsividis, 1982; Nay and Budak, 1983; Han and Park, 1984; Tsividis, 1986; Nagaraj, 1986; Singh et al., 1989; Wang, 1990 a,b ; Wang and Guggenbuhl, 1990; Vavelidis and Tsividis, 1992, 1993; Sakurai and Ismail, 1993; Takagi et al., 1994; Sellami, 1997; Vavelidis et al., 1997; Osa et al., 1998; Papazoglou and Karybakas, 1999; Tran and Wilamowski, 2001; Ramirez-Angulo et al., 2005; Popa, 2007; Wee and Sarpeshkar, 2008). Unfortunately, circuits emulating linear resistors can comprise ten and more transistors while lower transistor count circuits that emulate resistors tend to suffer from nonlinear effects. Furthermore, some MOS pseudo-resistors do not work over the full voltage range of a fabrication process and are limited to smaller voltage ranges (in the order of less than 2V in a 3.3V power supply process).

5.2 Mismatch in analog VLSI circuits

Copies of the same device or circuit on the same chip exhibits differences in their properties. These differences in properties are generated by slight variations in the fabrication process. The result is that if no compensation techniques are applied two circuits having the same layout will generate different output signals for the same input signals. Fabrication mismatch can become a major challenge for the design of optic flow sensors that are discussed in this thesis.

This section discusses the impact and causes of device mismatch in aVLSI (section 5.2.1) and shows strategies for mismatch compensation and reduction (section 5.2.2).

5.2.1 Impact, classification and causes of device mismatch in aVLSI

Device mismatch can cause implementations of optic flow algorithms in aVLSI technology to fail. Since the impact of device mismatch is not typically included in circuit simulations (unless the designer carries out more simulations for e.g. using the Monte-Carlo method), an implemented algorithm that works correctly in simulation can fail when tested on the physical chip. Many of the optic flow algorithms that I discuss in chapter 6 assume that all pixels of an optic flow sensor show the same behavior. With

the presence of mismatch, these algorithms can - depending on the amount of circuit mismatch - provide measurements that are not useable.

Mismatch is not only a problem for the circuits that compute optic flow but also for those circuits that implement the phototransduction or imaging process where mismatch is known as the so-called fixed-pattern noise (FPN) . In imagers FPN is typically mainly caused by the mismatch of the local readout transistors in each pixel (Fry et al., 1970).

Mismatch models that allow the prediction of mismatch before fabrication can be found in the literature (Shyu et al., 1982, 1984; Lakshmikumar et al., 1986; Pelgrom et al., 1989; Forti, 1994; Pavasovic et al., 1994; Minch et al., 1996; Bastos et al., 1997; Serrano-Gotarredona and Linares-Barranco, 1999*a,b*, 2000). The results of these investigations show that there are four different **categories of mismatch**:

1. **The striation effect** is a mismatch type that shows itself as a low-frequency sinusoidal variation of transistors properties in space.
2. **The gradient effect** is like the striation effect a spatial variation of device properties but of even lower frequency.
3. **Random variations** due to changes in transistor length, width, and other transistor parameters during fabrication. These random effects are inversely proportional to the square root of the transistors' area (Pavasovic et al., 1994).
4. **Edge effects** form a class of mismatch that caused by differences in the edges of a device. A long and skinny transistor might be less affected by random variations due to its large area. However the small width still makes it susceptible to mismatch caused by deviations in the fabrication of the edges of its gate along.

To understand the **causes for device mismatch** one has to remember that the feature sizes of these devices are only in the order of nanometers to micrometers. These devices are fabricated during a process containing several steps of photolithography, etching, doping - a process used to form the nFETs and pFETs where impurities are intentionally introduced into the silicon substrate through diffusion or ion implantation -, and deposition. Since none of these process steps can be perfectly controlled at sub-micrometer scales and since the fabricated structures undergo stress that can create slight deformations and changes in material properties, imperfections in the fabricated devices cannot be completely avoided. Furthermore, after fabrications due to parasitic effects the elements influence each other through undesired electric coupling effects such as parasitic capacitances at each node. Not all causes for device mismatch are understood to an extent that would allow their precise prediction.

5.2.2 Mismatch compensation and avoidance in aVLSI

Designers use various strategies to minimize these mismatch errors in VLSI circuits. These strategies can be divided into two classes: static and dynamic approaches.

1. **Static approaches** try to minimize device mismatch by applying proper layout rules during design. The mismatch effects can also be reduced through a one-time calibration of the circuits after fabrication. These values are stored in an off-chip look-up table and are subtracted from the pixel values during chip operation.
2. **Dynamic approaches** try to remove mismatch during operation by constantly running feedback loops or by periodically interrupting normal device operation to measure and store mismatch error values like offset errors or they try to avoid mismatch by reusing one and the same device for different operations. A general disadvantage of dynamic approaches is that they require the generation of a clock signal that can cause additional noise. Furthermore, dynamic approaches can limit the bandwidth of circuits. On the other hand they allow for constant adaptation to circuit variations.

In the following sections I introduce a variety of static and dynamic approaches for handling mismatch errors.

Layout considerations

There are several layout techniques that have been developed to reduce device mismatch. I only give five examples. Many more approaches can be found in standard textbooks.

1. The absolute value of the random mismatch error depends on the square root of the transistors' area (Pavasovic et al., 1994). So a static approach to reduce the effect of mismatch is to **increase the transistors' width and length**. However, since fabrication costs are closely linked to the layout area occupied by the circuitry, this approach is costly especially for high-volume chip production.
2. To improve matching between devices that can be closely placed together, there are additional layout techniques. If for example the transistors of a differential pair should be well-matched, it is advantageous to **place these transistors close** to each other using **the same orientation** (Liu et al., 2002) so that scatter effects

during fabrication affect both transistors in a similar way. This approach is clearly limited to circuits whose matched devices that can be placed closely in space. This approach is not suitable for circuits whose transistors are distributed over the whole chip like the pixel array of an optic flow sensor.

3. If one device should for example have twice the width of another device it is typically wise to create **unity-size devices** and make the bigger device from two of that unity-size devices. Although this leads to an increase of layout space, that way one can make sure that the one device really has twice the width of the other.
4. To avoid edge and surround effects the chip designer should try to make the surround of the devices the same by placing so-called **dummy devices**. A dummy device is a device that has the same shape and properties as the devices in the array. Dummy devices can be single transistors or capacitors but also complete pixels. For example the pixel array of my linear optic flow sensors is surrounded by dummy pixels that have the same layout as the normal pixels but that are not used as part of the optic flow computation. That way I ensure that all functionally used pixels are surrounded by neighboring pixels and suffer less from edge effects.
5. Finally, better device matching can also be achieved by a proper **choice of** the fabrication process and **material** for the implementation of a device. As mentioned earlier, poly1-to-poly2 capacitors have better matching than MOS capacitors.

Proper layout considerations are usually the minimum of what should be considered to reduce device mismatch.

Calibration with off-chip lookup tables

One approach for chip calibration is to measure the chip's outputs \hat{O} for a test set of inputs I one time after fabrication. The measured outputs \hat{O} can then be stored together with the desired mismatch-free outputs O in a static lookup table. A lookup table is a data structure that is widely used in computer science to replace a complex runtime computation by simply retrieving a value from memory.

In theory, after building the lookup table the corrected desired chip output can be retrieved while operating the chip by searching for the stored output value O in the lookup table for the measured error-prone chip output \hat{O} . Practically, to limit the number of data sets in the table one often cannot store values of O for all possible chip outputs

\hat{O} . Thus one would have to either find the closest entry for \hat{O} or to perform some linear interpolation between the closest entries.

The following conditions have to be fulfilled so that a lookup table can be applied for mismatch compensation:

1. The lookup table approach is limited to bijective transformations between the measured chip outputs \hat{O} and desired error-free output value O .
2. \hat{O} is only allowed to be dependent on the current chip input I and not on previous inputs or an internally unknown chip state.

Since the chip can usually not be operated during the calibration process the lookup table is typically generated only once. However, if the chip changes its behavior over time, for eg from aging, the calibration procedure has to be repeated periodically. In any case the creation of the lookup table can be a time-consuming process when many $\hat{O} - O$ pairs need to be tested. On the other hand if the chip's output behavior is not too complex using a lookup table for mismatch compensation is probably the technique with the lowest overhead since it does not require additional on-chip circuitry.

Look-up tables are widely used for calibration of analog VLSI circuits. If the analog circuits are integrated together with a microprocessor and the necessary analog-to-digital converters in a so-called system-on-chip (SoC) on the same chip, the lookup table can be stored in the flash memory or EPROM of that microprocessor and the calibration can even be performed on-chip.

I use external look-up tables for the calibration of the dynamic optic flow sensor described in chapter 7.

Mismatch compensation with on-chip nonvolatile memories

Another approach to compensate for device mismatch in aVLSI technology is the application of on-chip nonvolatile memories like those available through floating gate technology. The goal of this technique is to cancel device mismatch by trimming the circuit's behavior after fabrication and storing mismatch compensation values on-chip.

In the 1980s Alspector and Allen (1987) suggested the use of floating-gate devices for storing the analog weights of neural network devices. Floating-gate memories were also used for storing analog weights in learning arrays (Diorio et al., 1997), for cancellation of fixed pattern noise and compensation of background illumination in CMOS imagers (Aslam-Siddiqi et al., 1998), for trimming digital-to-analog converters

(Figueroa et al., 2001), for auto-zeroing input offsets in operational amplifiers (Hasler et al., 2001) and for on-chip compensation of mismatch effects in analog VLSI neural networks (Figueroa et al., 2004). We have used floating-gate memories for the calibration and nonvolatile storage of analog synaptic weights of aVLSI neural network circuits (Moeckel, 2005; Liu and Moeckel, 2008).

On-chip nonvolatile memories have the advantage that they allow for local compensation of mismatch errors. This local compensation has the advantage that the errors do not combine within the path of computation in a way that would make them impossible to be removed afterwards because the precondition of a bijective transformation between error-prone and error-free output values is not fulfilled after the combination of multiple errors.

The on-chip storage of compensation values can be implemented through memory transistors that comprise a floating gate - a gate that is completely surrounded by an insulator ensuring that no charge can leave the transistor's gate during normal operation of the transistor. The charge stored on the floating gate is updated taking advantage of the two physical processes "Fowler-Nordheim tunneling" (Lenzinger and Snow, 1969) and "Impact ionized hot-electron injection" (Sanchez and DeMassa, 1991).

Floating gate memory transistors can be implemented in standard CMOS technology but they require some overhead: Both Fowler-Nordheim tunneling and impact ionized hot-electron injection require supply voltages that are above those usually used during normal operation of the circuits. Providing and working with these high voltages requires great care during the design of the chip's layout. The necessary infrastructure to control the tunneling and injection process typically increases the required chip area and thus reduces the fill factor of an optic sensor but it also causes additional overhead in terms of off-chip electronics to generate the additional power supplies and control signals. This overhead is the reason why we decided not to make use of floating-gate transistors for the optic flow sensors and imagers presented in this thesis.

Correlated double sampling

Correlated double sampling (CDS) is a popular approach for the removal of undesired offsets. CDS techniques are widely used in imagers but also in optic flow sensors for removing fixed-pattern noise (FPN) (Mehta and Etienne-Cummings, 2003, 2004; Gruev and Etienne-Cummings, 2004; Yang et al., 2006; Han and Yoon, 2006) as well as for the removal of offset errors in operational amplifiers (Enz and Temes, 1996). The idea behind CDS is to measure the output of a sensor twice in the same read cycle: First,

the pixel output is measured right after reset. Then a second measurement after an integration period of the pixel. The mismatch due to the readout transistor in the pixel can be removed by subtracting the two measurements.

The fixed-pattern noise in active pixel imagers are mainly caused by the mismatch between the local readout transistors of the individual pixels (Fry et al., 1970). One can remove these offset errors through correlated double sampling as discussed above.

The circuit **overhead** for the CDS scheme in an active pixel imager is relatively low. Since all pixels have to be reset anyway every time before a new image is taken, the overall overhead is only the CDS circuit that does the subtraction of V_{OUT1} and V_{OUT2} as well as one single analog memory that stores V_{OUT1} of the pixel that is currently read while V_{OUT2} is measured. If the pixels are read sequentially only one CDS circuit has to be implemented for each column of a pixel array thus making this technique very attractive for the removal of fixed-pattern noise in imagers and optic flow sensors. One disadvantage of CDS lies in the fact that the whole pixel array needs to be read twice. This reduces the imager's bandwidth but does not need to cause a substantial decrease of the frame rate when the readout time is much smaller than the exposure time - the time that the pixels need to take an image.

Beside the particular voltage-mode CDS scheme described in the previous example, other voltage-mode and current-mode CDS techniques have been described each having their particular advantages and disadvantages. See the publication by Yang et al. (2006) for a comparison between voltage and current-mode CDS schemes in aVLSI CMOS imagers.

Time multiplexing

Sharing circuits through time multiplexing is another approach to avoid mismatch. The idea behind time multiplexing is that if the circuits for two operations have to be well-matched, then one could simply reuse one and the same circuit to perform the operations one after another. Since the same circuit is reused, the operations should not be affected by circuit mismatch.

Time multiplexing techniques come with the additional requirement for switches, memory circuits and clock signals. However, they have the advantage that they allow to trade off the circuit's bandwidth against the chip area. If circuits that require a lot of layout space are necessary on-chip, it can be advantageous to use only one such circuit and to share this circuit by multiplexing in time at the expense of additional overhead in the control circuitry.

5.3 Noise

Besides mismatch, circuit noise has main impact on the quality of the outputs of any aVLSI sensor. In comparison to mismatch, noise leads to variations in the current and voltage outputs not only between two circuits of the same type but also in the output of the very same circuit. Noise affects the outputs of a circuit in an unpredictable way even when the inputs to the circuit do not change. There are different types of noise and ways to deal with them. Some examples of noise that can be found in aVLSI circuits are:

- **Thermal noise** can be observed in any conducting material including MOSFETs and resistors. It is generated by thermal agitation of the charge carriers inside an electrical conductor at equilibrium (Wikipedia, 2012a). Thermal noise is generated even if no voltage is applied to the conductor. In an ideal resistor thermal noise is approximately white. The root mean square voltage due to thermal noise v_n , generated in a resistance R over bandwidth Δf , is given by

$$v_n = \sqrt{4k_B T R \Delta f} \quad (5.13)$$

where k_B is Boltzmann's constant and T is the resistor's absolute temperature in kelvin (Wikipedia, 2012a). The thermal noise of an RC filter is given by

$$v_n = \sqrt{k_B T / C} \quad (5.14)$$

where C is the capacitance of the RC filter. Interestingly, the value of the resistance is removed in (5.14) since it is canceled when substituting the bandwidth in (5.13) by the expression for the RC circuit - $\Delta f_{RC} = 1/(4RC)$. This allows to reduce the effect of thermal noise in a circuit by adding a sufficiently large capacitor.

- **Shot noise** is a form of noise that is explained by the discrete nature of electric charge carriers and photons (Wikipedia, 2012b). When light is collected by the photodiode of an optical sensor, photons generate charge carriers in the photodiode. If the number of photons per unit time is relatively small the current formed by the charge carriers appears to be noisy following a Poisson distribution. This is the case when the optical sensor is operated at low light intensities. At higher intensities due to the large number of photons being collected the shot noise follows

a Gaussian distribution. Similarly as thermal noise, shot noise can be observed in any conductor but it requires a voltage to be applied that generates a stream of electrically charged carriers.

- Any two conducting materials that are separated by an insulator form a (parasitic) capacitor on an aVLSI chip. For examples two metal wires running on top of each other or any metal wire running on top of the gate of an MOSFET form a capacitor. If one electrode of a parasitic capacitor carries a signal of high frequency and amplitude the signal on the other electrode will be affected by **cross talk due to capacitive coupling**. To avoid coupling effects a chip designer typically tries to minimize the area of overlapping wires and gates. Furthermore, separating digital circuits and analog circuits has proven to be effective to reduce coupling effects. Finally one should use low resistance metal wires connected to the power supply to shield against capacitive couplings.
- Analog VLSI chips like optic flow sensors are typically designed with the assumption that the power supply of the circuits remains constant and noise-free. **Noise on the power supplies** has major impact on all circuits since it affects the operation of all MOSFETs. So special care has to be taken to supply all MOSFETs on the chip with power using wires of low resistance. Furthermore, it is common practice to split the power supply of MOSFETs being operated as digital switches and those used for analog circuits and to place filters off-chip on the printed circuit board that carries the chip. Nevertheless, a certain amount of noise on the power supply - typically a few millivolt on a 3.3V supply - cannot be avoided.

In summary, special care has to be taken so that the implementation of an optic flow algorithm in aVLSI technology does not fail because of circuit noise. Similarly as for mismatch a circuit designer can apply several techniques to minimize the effect of circuit noise but it is of great benefit if the chosen algorithm already shows some robustness against noise.

5.4 Conclusion

This chapter gives a short introduction into the properties of basic devices in analog Very Large Scale Integrated circuits. The chapter highlights the challenges for the design of aVLSI chips such as device mismatch and circuit noise. Both effects can

prevent a successful implementation of an optical flow algorithm into aVLSI technology. The chapter describes several design techniques for dealing with these challenges.

Chapter 6

Evaluation of optical flow sensors for autonomous robots

Abstract

This chapter presents evaluation criteria for VLSI optical flow sensors and algorithms for autonomous robotic platforms - especially micro aerial vehicles (MAVs). Such platforms require small, light-weight, and low power sensor modules and electronics and provide limited computational power for processing sensory signals in real-time. This chapter also presents an evaluation of existing VLSI optical flow sensors and their implemented optical flow algorithms to identify promising candidates of optical flow algorithms for VLSI sensors that can be used for micro aerial vehicles.

6.1 Sensor optics, pixel size, and fill-factor

The human eye can constantly adjust the focal length of its lens as well as the size of its aperture and thus the distance at which objects are optimally projected onto its photoreceptors in form of a sharp image.

In contrast to the human eye, bees and flies cannot vary the aperture and focal length of their compound eyes. MAVs typically share this fixed optics constraints with insects since they as well can only carry light weight optics which are usually of fixed focal length. For these MAVs the distance at which the best quality optical flow is generated has to be set before flight by selecting a lens that suits the optical flow sensor, the flight controller and the MAVs flight properties. The MAVs controller and flight properties

determine the object distance at which an MAV has to turn away from an object to avoid a collision. Around this distance the optical flow has to be measured with best reliability to avoid collisions with obstacles.

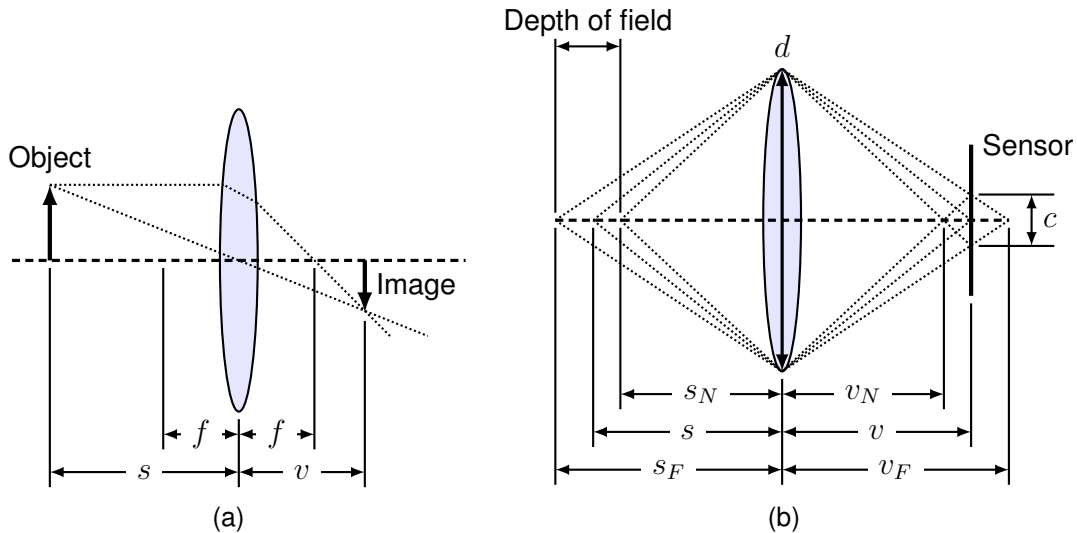


Figure 6.1: (a) A thin lens generates a real image at the distance v that is a projection of an object at the distance s . In this lens model the focal length of the lens, f , is the main parameter that determines at which distance the image is sharply projected. (b) A lens with fixed focal length f can only optimally focus objects at one fixed distance s . Points closer or further away from the lens will not be projected in form of a sharp image point but as a blurred spot with a diameter c - the so-called circle of confusion. The depth of field gives the difference between the maximum (S_F) and minimum (S_N) object distance that still leads to an acceptable c . The aperture d of a lens allows regulating the depth of field for a lens with fixed focal length.

The distance where the sharpest image is generated on the surface of the sensor is set by the **focal length** f of a lens. Figure 6.1a depicts the generation of an image from an object by a thin lens. According to the well-known model of a thin lens a sharp image is generated at the distance v from an object at distance s according to the law

$$\frac{1}{s} + \frac{1}{v} = \frac{1}{f} \quad (6.1)$$

from which can be concluded that all objects at a distance s much greater than f will lead to images that are sharply projected at the distance $v = f$.

A lens with a fixed focal length can only optimally focus objects at one fixed distance while the image sharpness gradually decreases when objects at this optimal distance move closer or further away from the lens (see figure 6.1b). The **depth of field** gives the distance between the nearest (at distance S_N) and farthest (at distance S_F) objects that still produces an acceptably sharp image. The depth of field of a visual sensor system is influenced by the focal length, and **f-number** , F , of the lens

$$F = \frac{f}{d} \quad (6.2)$$

- which decreases with the size of the lens' aperture d increasing. The depth of field furthermore depends on the maximum circle of confusion or image blur c that should be allowed. A large depth of field is of advantage when objects at a large range of distances should generate optical flow on the vision sensor. Under constant-speed MAV control and in a stationary environment, a large depth of field is often not required nor desired. Objects at distances that are too close to be avoided do not need to be detected. Objects that are far away need not be detected so as to simplify the MAV's controller that should deal only with objects that are behaviorally relevant. **MAVs that vary their velocities** or that do not fly in stationary environments require a larger depth-of-field since the distance at which the MAV has to turn away from an obstacle varies with the relative speed between the MAV and the obstacle that should be avoided.

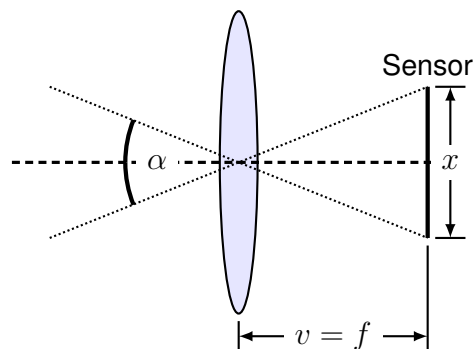


Figure 6.2: The field of view of a sensor pixel or entire sensor $FOV = \alpha$ depends on the focal length of the lens as well as on the pixel or sensor size.

Besides the depth of field and optimal focal distance, the focal length of a lens also influences the **field of view** of a vision sensor system (illustrated in figure 6.2). The field of view can be defined both for a single pixel and the entire sensor area covered with photoreceptors. The field of view of a single pixel sets the minimum size of a texture feature that can still be resolved. The field of view of the entire sensor determines the

maximum angle at which objects are still - but not necessarily sharply - projected onto the surface of the photosensing area of the vision sensor. Since in MAV control the focal length is typically set by other requirements, the field of view can only be further adjusted by the sensor area covered with photoreceptors and the size of the individual photoreceptors. The field of view, FOV , of a single pixel or entire sensor is

$$FOV = \alpha = 2 \cdot \arctan \frac{x}{2f} \quad (6.3)$$

where x is the size of the pixel or of the photosensing area of the sensor, respectively.

The **weight of a vision sensor module** increases with the diameter and focal length of the lens. Since the distance between the vision sensor and the lens should be about the focal length to allow sharp projections onto the sensor's surface, a lens with a larger focal length requires as well a larger and thus heavier mechanical lens holder. As a result on MAVs the focal length cannot be chosen freely but is restricted also by the maximum payload the MAV can carry. Only the lens' aperture remains to set the depth of field and thus the distances at which objects can still be detected. The weight of the sensor module - the combination of vision sensor, lens and lens holder - furthermore depends on the material of the lens. Typically for MAVs light weight plastic lenses are preferred.

To allow the creation of a small and light weight optical flow sensor module for the airplane by Zufferey et al. (2007) (see section 4.1 for a detailed description) a plastic lens with a focal length of 2.43mm was chosen for the adaptive vision sensor presented in this thesis (for detailed lens and sensor specification see section 9.1). The choice was also motivated by the lens' weight of 0.18g that led to a total weight for an optical flow sensor module of about 0.3g.

For the desired properties of the optical flow or vision sensors suitable for flight control of MAVs the restriction to lenses with small focal length and diameter lead to the fact that sensors with **small pixel size** and **high fill-factor** (ratio of area occupied by the photodiode to the total pixel area) are required to resolve small texture features. Sensors with larger pixels require lenses with larger focal length to achieve a small field of view for single pixels (which is essential to resolve small texture features) and lenses with larger diameter to gain the same total field of view like a sensor with small pixel size. Thus sensors with big pixel size lead to heavier sensor modules not only because of the increased sensor size but also because they require larger lenses and lens holders. A large pixel field of view is not acceptable since the spatial resolution is a key feature of optical sensor modules for allowing MAVs to leave well-controlled lab envi-

ronments where walls and objects are covered with artificial high contrast patterns with low spatial frequencies.

With the chosen lens the adaptive vision sensor presented in chapter 9 has a pixel field of view of about 0.4° . The sensor covers a field of view of about 66.7° .

The pixel size requirements impact the choice of the sensor architecture and optical flow algorithm. Since the circuits on a sensor are distributed on a 2-dimensional plane, the implementation of complex optical flow algorithms can result in pixels with a small fill-factor and large total area which then limits the number of pixels that can be placed in a fixed chip area. Optical flow sensors found in the literature typical show pixel fill-factor that vary between 2.5% to 10%.

Figure 6.3 depicts several options to increase a sensor's spatial resolution for pixels with a low fill factor at different costs by rearranging the sensor's layout:

1. Pixels can be arranged in form of a linear array with the photosensing area of neighboring pixels placed close to each other and the processing circuits be placed at the side of the photosensing area (figure 6.3b). In this arrangement optical flow outputs can only be computed in one dimension along the array of photoreceptors.
2. Optical flow in two dimensions can be measured when pixels are arranged in form of two linear arrays that face each other as shown in figure 6.3c. However, in this configuration the sensor's field of view in the direction orthogonal to the orientation of the linear array remains small in comparison to a regular 2-dimensional pixel array layout as the one shown in figure 6.3a.
3. Finally, the circuits for optical flow calculation can be spatially separated from the photoreceptor stage as depicted in figure 6.3d. This way only circuits for the read-out of the photoreceptors need to be placed inside the photoreceptor array. The result can be a classical imager spatially separated from the processing circuits that are still placed on the same sensor. A good spatial resolution and estimation of optical flow components in two dimensions can be achieved. However, this layout technique either leads to irregular pixel layouts that are more vulnerable to circuit noise and mismatch when the photoreceptors have to be connected to the circuits for optical flow calculation with individual wires. If the data from individual pixels is transmitted to the circuits outside the photoreceptor array via a bus system mismatch effects can be better controlled but a reduction of temporal resolution has to be accepted since the photoreceptor outputs have to be transmitted sequentially via the bus system.

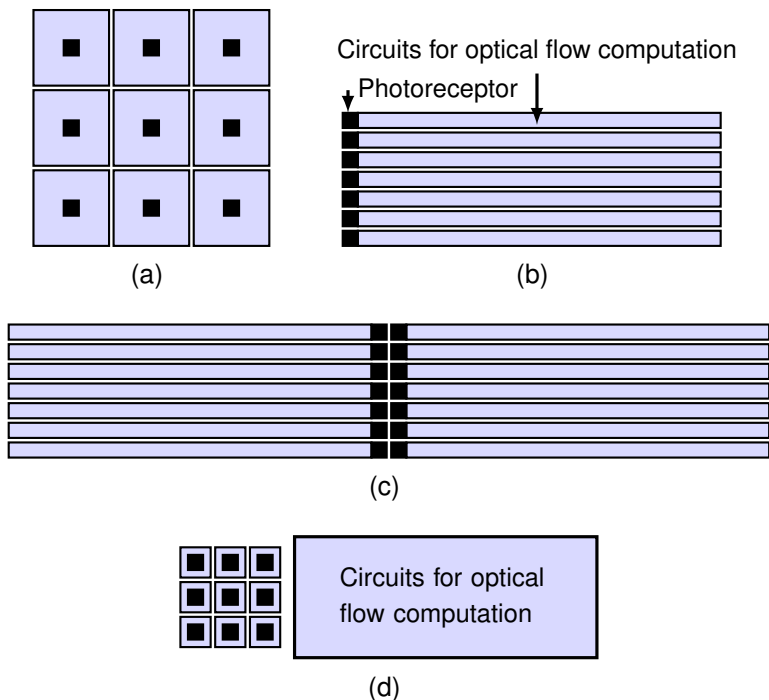


Figure 6.3: By changing the arrangement of photoreceptors and circuits for optical flow computation on the focal plane a sensor's spatial resolution can be adjusted while keeping the fill factor of the pixels constant. Pixels can be organized (a) in form of a regular 2-dimensional array, (b) a linear array that is only sensitive to optical flow in one direction, (c) in form of two 1-dimensional arrays where photoreceptors are facing each other or (d) the circuits for optical flow computation can be separated from the photoreceptor array. (a-c) Fill-factor equals 16%. (d) If the circuits for optical flow computation are placed on the side of the photoreceptor array and data is sequentially read from the photoreceptors, part of the circuits for optical flow computations can be shared and the total sensor area can be reduced.

Option 1 was selected for the design of the optical flow sensors presented in this thesis (chapters 7 and 9). As discussed in section 4.4 the restriction that with a linear photoreceptor arrangement optical flow can only be measured in one dimension is acceptable for the reference MAV by Zufferey et al. (2007).

6.2 Evaluation criteria for optical flow algorithms and sensors

As presented in chapter 4, there is a variety of flying platforms including airplanes, helicopters, quadcopters and flappers each having different flight properties that require different controllers. Different sensor arrangements include minimalistic configurations with only one or two sensors and a medium field of view of 30° like presented by Zufferey et al. (2007) (section 4.1), configurations that use an array of several sensors each with a small field of view (see Beyeler et al. (2009) or section 4.2) and arrangements that use only a single sensor in combination with a fish-eye lens that can provide a field of view of up to 360° . Each sensor arrangement requires a different controller and different specifications of the optical flow sensors. Also the MAV's flight controller and flight properties affect the required properties of the sensors. For successful autonomous flight control, the optical flow algorithm, its implementation as part of the VLSI sensor, the optics, MAV flight properties as well as the flight controller need to be chosen so that all components work together efficiently and effectively.

The variety of possible MAV properties, flight controllers, and sensors arrangements make it difficult to find a general metric showing the minimum requirements for an optical flow sensor in addition to the criteria that arise from the limited choice of optics (see section 6.1). Still this section provides a list of properties that an ideal optical flow sensor system should have:

Robustness and reliability of optical flow outputs

To be useful for MAV control VLSI optical flow sensor should provide robust and reliable optical flow outputs over a large range of stimulus velocities. Optical flow outputs must be independent of background light intensity and stimulus properties.

- The range of **background light intensities** an optical flow sensor is confronted with is set by the environment in which the MAV should be operated. In indoors environments, the background lighting can be well controlled so that the sensor does not need to adapt to a large range of lighting intensities. In outdoor environments however light intensities can vary from 10^{-4} lux at starlight, to 0.27 lux at full moon, to 300 – 500 lux office lightning, to 10,000 – 25,000 lux daylight, to up to 130,000 lux when sunlight is directly shining on a surface. The human eye is capable of working under all these conditions - most optical flow sensors are not.

- Providing a reliable optical flow output independent of the **contrast of the visual stimuli** is challenging but crucial for optical flow sensors. The challenge is that stimuli with low spatial contrast typically generate signals with small amplitude that suffer most from circuit noise.
- All physical circuit implementations suffer from parasitic capacitance and resistance effects that lead to a limited bandwidth of these circuits and thus of a limitation of **temporal frequencies** of the signals that can be processed. Care has to be taken during the circuit design to avoid unwanted limitations.
- **Circuit noise** (discussed in more detail in section 5.3) corrupts the optical flow outputs of a sensor. So algorithms and their implementations that suffer less from noise and those that support the reduction of noise should be preferred.
- Analog circuit sensor designers have to tackle the effect of **device mismatch** on the optical flow pixel responses. Unlike computer simulations, any two pixels on the same chip will not produce exactly the same analog output response to the same stimulus. As presented in section 5.2 there are different circuit techniques to reduce the effect of device mismatch. Algorithms that do not expect the same flow output from each pixel are of advantage because they are less susceptible to device mismatch when implemented in aVLSI optical flow sensors.

Besides having access to reliable and robust optical flow output signals, a flight controller can benefit from the on-chip generation of a **reliability signal**. This signal can, for example, indicate the number of visual features that have been detected or the contrast of the stimulus that generated the current flow signal. The reliability signal gives information about the quality of the optical flow signal. This information is useful for the further processing of the optical flow signals since it allows a flight controller to identify outliers and to ignore a set of optical flow measurements that are not reliable.

Spatial integration and motion segmentation

Besides circuit noise (discussed in section 5.3) the aperture problem leads to noisy and incorrect optical flow outputs of individual sensor pixels. The **aperture problem** arises from the fact that the individual pixels of a sensor have only a limited field of view leading to an ambiguity in determining the two velocity components of an optical flow vector. **Spatial integration** of optical flow outputs from individual pixels allows for a correction of individual optical flow outputs. Spatial optical flow integration can

act as a spatial low pass filter that reduces circuit noise. It furthermore can help to generate the correct optical flow values at pixels that are confronted with stimuli of too low contrast and thus cannot compute a local optical flow output without information from other pixels.

Spatial integration does not just mean averaging the local optical flow outputs of individual pixels. For example, a measured optical flow value of zero optical flow can mean either the relative velocity of the stimulus is zero or that there is no stimulus or that the stimulus contrast was too low to generate a finite flow value. A good integration algorithm has to distinguish between these cases and for instance weights the individual optical flow outputs according to the contrast of the stimuli that generated the optical flow. The addition of a smoothness constraint for optical flow estimation has for example been proposed by Horn and Schunk (1981). VLSI sensors that implement spatial integration have been presented by Tanner and Mead (1986); Stocker and Douglas (1999) and Stocker (2002).

Algorithms for spatial integration and motion segmentation can be complex and typically require additional circuits on the optical flow sensor - thus increasing the sensor's pixel size and making it more vulnerable to circuit mismatch. This is why the optical flow sensors presented in this thesis do not include on-chip mechanisms for spatial integration and motion segmentation. For calculating optical flow from the adaptive vision sensor (chapter 9) we use the image interpolation algorithm by Srinivasan (1993, 1994). As presented in section 9.3 this algorithm implements spatial integration but is currently executed on a microcontroller.

When flying in natural environments **motion segmentation** can be used to distinguish different objects in the surround. Because translational optical flow is dependent on the distance between the sensor and objects in the environment (see observation 2 in section 2.3) objects at different distances will generate different flow fields on the sensor. Thus abrupt changes in the measured flow field indicate the boundaries of objects. If spatial integration is applied across these boundaries corrupted optical flow outputs can be generated. This is why Stocker (2002) integrated thresholds into their VLSI sensor that allow to break the smoothing across local optical flow outputs at motion boundaries.

6.3 Evaluation of algorithms for optical flow sensors and their VLSI counterpart

This section expounds on the various algorithms implemented on the analog VLSI optical flow sensors and the trade-offs that have to be made in these implementations. The presented analysis has been published as part of a book chapter (Moeckel and Liu, 2009).

Challenges of evaluation

The comparison of existing optical flow sensors in the literature is challenging because of the following reasons:

- In scientific publications, the novelties of a sensor design are emphasized but **not all chip specifications are given** as it would be the case for commercial sensor that come with datasheets. This makes a judgment that includes all my evaluation criteria difficult and often impossible. For commercially available sensors on the other hand **details of the implemented optical flow algorithm** are typically **not given** because of proprietary information.
- The **same optical flow algorithm can be implemented with very different circuits**. The state and value of a variable for example can be encoded as analog voltage, analog current, analog time delay or as a digital number.
- As shown in section 6.1 **pixel form and arrangement** have a major impact on sensor properties like spatial resolution.
- Finally, the available VLSI **fabrication processes** have evolved over time. Fabrication processes highly influence sensor properties including the power consumption, size and bandwidth of single transistors, efficiency of photodiodes and the types of available capacitors. The result is that sensor parameters like fill factor, pixel size, power consumption, the minimum and maximum spatial frequency and velocity of detectable stimuli as well as the minimum amount of light that a photoreceptor requires to operate reliably can vary with the specific process in which the circuits have been fabricated.

Because of the different circuit techniques used by the individual circuit designers and the variety of fabrication processes for the reported optical flow sensors, I have refrained

6.3. Evaluation of algorithms for optical flow sensors and their VLSI counterpart⁸³

from making explicit comparisons between the different implementations in figure 6.4. I concentrate instead on a comparison between the implemented optical flow algorithms and show their corresponding limitations in the implemented aVLSI form.

Categories of VLSI optical flow sensors

Since the first aVLSI motion chip which was designed for use in the Xerox optical mouse in 1981 by Lyon (1981), more than 50 different motion chip implementations have been published. Some major contributions are shown in the timeline chart of figure 6.4. For my evaluation I focus on optical flow sensors that implement both the phototransduction circuits as well as the circuits for optical flow computation on a single chip. Solutions that combine a vision sensor together with a separate device for optical flow computation are not included in the survey. The list of sensors in figure 6.4 is not meant to be complete. Figure 6.4 reflects my personal view on which single-chip optical flow sensors fall in a specific category.

As shown in Figure 6.5, the algorithms implemented on-chip can be divided into two primary groups: (1) intensity-based algorithms and (2) token-based algorithms.

- **Intensity-based algorithms** estimate optical flow directly from the image brightness. Optical flow is estimated based on either (1a) the gradient of the image brightness or (1b) the correlation of the image brightness of neighboring pixels. The image brightness in the implementations could be the equivalent of the output of some pre-processing on the visual input: for example, the absolute intensity, or the log intensity.
- **Token-based algorithms** estimate optical flow by tracking detected features or tokens across space and time. The term *token* comes from the field of computer vision where tokens are defined either as low-level features like edges or corners or high-level features like objects Ullman (1981). On the optical flow sensors, the token often consist of a binary pulse which is generated when the contrast of an edge exceeds a threshold. These algorithms can be further divided into two major sub-categories: (2a) Optical flow is estimated through the correlation of edge patterns. (2b) Optical flow is estimated by measuring the time taken for a feature to travel between two adjacent pixels. The latter are also called *time-of-travel* algorithms.

Intensity-based versus token-based: The advantage of intensity-based algorithms is that the motion computation is continuous whereas token-based algorithms update

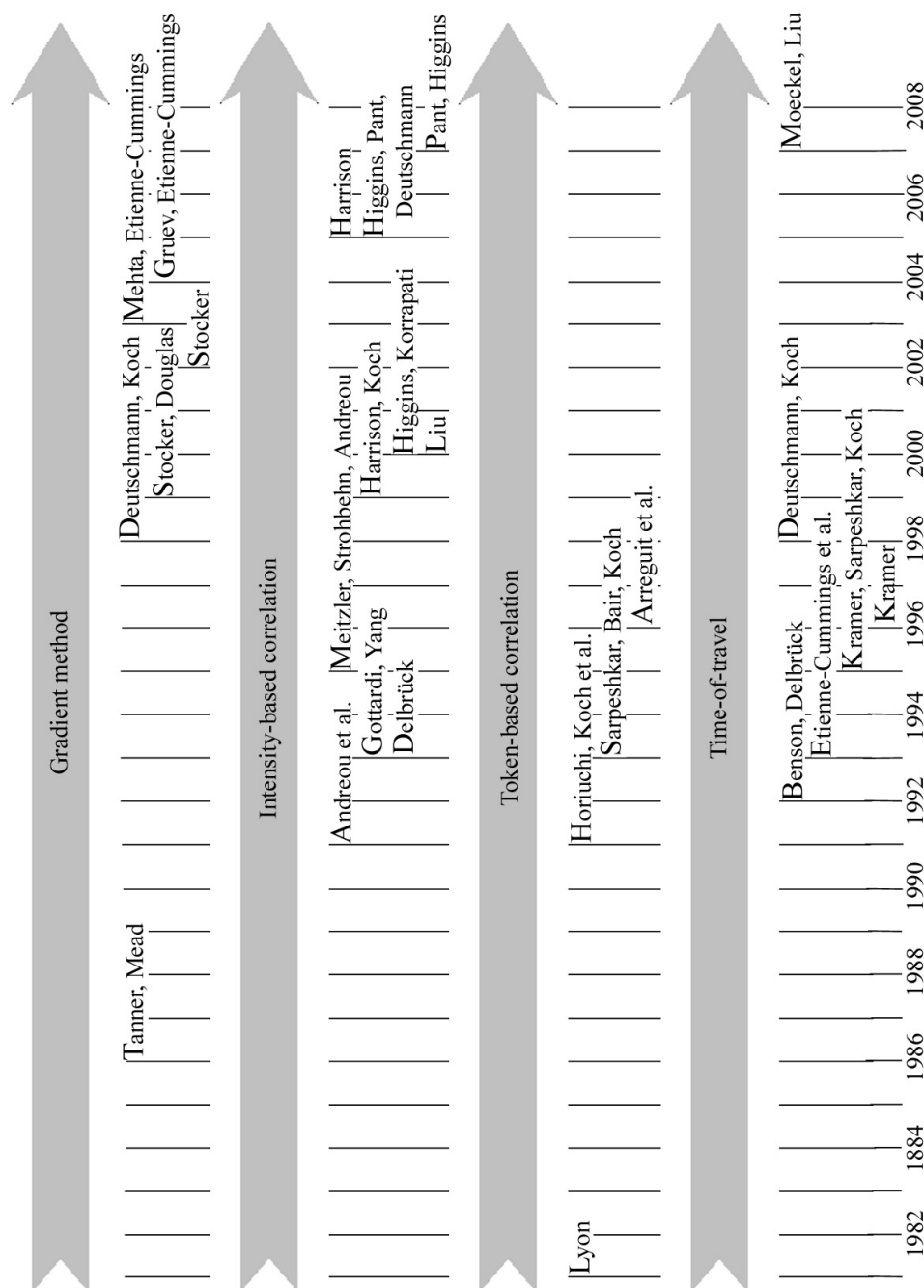


Figure 6.4: Timeline of single-chip VLSI optical flow sensor implementations. I show only major contributions of different authors. For example, publications based on one design are represented using the date of the first publication.

6.3. Evaluation of algorithms for optical flow sensors and their VLSI counterpart⁸⁵

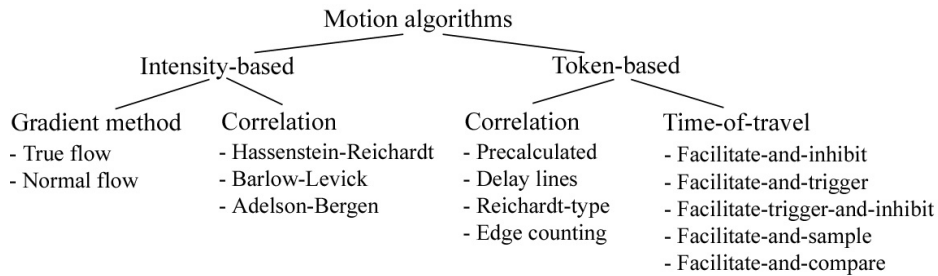


Figure 6.5: Overview of motion algorithms implemented in aVLSI.

the motion values only if a pre-defined feature is detected in the image. The different update dynamics mean that the host system which regularly samples the motion values will always receive the instantaneous motion value from an intensity-based motion chip but will most likely receive a motion value that was computed sometime in the past from a token-based motion chip. However, the readout of token-based motion chips could be an advantage if the feature detection signal or token is used to signal the host system when a new value is computed.

Token-based algorithms can be more robust against noise since only reliable detected features are used to determine the local image speeds. Their disadvantage lies in the fact that the accuracy of the computed motion is determined by the accuracy of the feature detectors. For example, if a feature is detected in a particular pixel but not detected in the neighboring pixel, token-based algorithms will produce an incorrect motion value. Hence, the interpretation of the motion value by the host system is more challenging. The last updated motion value can either be allowed to decay slowly over time at a predefined rate, or is reset after a set time interval. Token-based algorithms also have the advantage during implementation that the algorithms allow easy partitioning into the phototransduction, feature detection and motion detection blocks. This partitioning can simplify the design of the subsequent aVLSI circuits since each circuit can be optimized for its particular task.

A different way of categorization is to separate sensor designs into the groups of **bio-inspired** and **classical industrial implementations of machine vision algorithms**.

- Many of the various optical flow sensors from the past 20 years are based on biological models of optical flow processing, for example, the optical flow estimate extracted by cells in the insect visual pathway Egelhaaf and Borst (1993a). These implementations often make use of the photoreceptor circuit by Delbrück and Mead (1994) that operates in continuous time.
- The implementation of these bio-inspired algorithms should be contrasted with

machine vision implementations of optical flow algorithms on many mobile platforms. The latter use primarily outputs of clocked frame-based imagers from which optical flow is extracted. The frame-based computation methods lead to optical flow outputs that are only available at discrete sampling times in contrast to the continuous-time analog motion outputs that are available from many bio-inspired optical flow sensors.

In the literature many hybrid solutions can be found that combine bio-inspired circuits with classical industrial implementations.

In this evaluation I categorized the solutions into intensity-based algorithms described in sections 6.3.1 and 6.3.2 and the token-based algorithms described in sections 6.3.3 and 6.3.4.

6.3.1 Gradient-based intensity algorithm

The gradient-based algorithm for computing optical flow was one of the first algorithms implemented on an aVLSI chip. This algorithm is also often used in the computer/machine vision community. The optical flow equation is derived from the brightness constancy assumption that the image brightness, I , at a point (x, y) and at a time t stays constant. Taking the derivative of I with respect to time leads to $\frac{d}{dt}I(x, y, t) = 0$. Using the chain rule for differentiation one receives an expression that relates the image flow components in the x - and y -direction $v_x = dx/dt$ and $v_y = dy/dt$ and the partial change of the brightness ∂I :

$$I_x v_x + I_y v_y + I_t = 0 \quad (6.4)$$

where $I_x = \partial I / \partial x$, $I_y = \partial I / \partial y$ and $I_t = \partial I / \partial t$.

Equation 6.4 is underconstrained since only one independent measure of the image brightness $I(x, y, t)$ at a point in time and space is available while the optical flow velocity has two components v_x and v_y . By combining estimated flows spatially across pixels, one can arrive at a unique solution of the equation. Two main implementations of optical flow models have been proposed in the literature:

True flow: Horn and Schunk (1981) proposed a smoothness constraint for the optical flow field such that the flow field varies smoothly across space. Using both constraints, the local optical flow vectors $v = (v_x, v_y)$ can be combined to arrive at an optimal solution

for the optical flow estimate by minimizing the cost function:

$$H_{OF}(v) = \sum_{ij} [(I_x v_x + I_y v_y + I_t)^2 + \rho((\Delta^x v_x)^2) + (\Delta^x v_y)^2 + \rho((\Delta^y v_x)^2) + (\Delta^y v_y)^2] \quad (6.5)$$

where the global constant ρ describes the amount of spatial smoothing, and Δ^x and Δ^y represent the discrete derivative operators in the x and y directions respectively for each i,j th pixel in the array.

The first optical flow sensor that implemented an energy minimization equation was proposed by Tanner and Mead (1986) with a subsequent higher performance version described by Stocker and Douglas (1999). The Tanner and Mead chip is one of the first aVLSI optical flow sensors and is often cited as the first single-chip optical flow sensor which did not depend on a specific visual pattern environment. The chip contained an array of 8 by 8 pixels and produced a global velocity value over a limited range of input contrasts. It did not implement the smoothness constraint term in (6.5). The chip by Stocker and Douglas (1999) implements this term and provides smooth optical flow values. However, the smoothness constraint term does not take into account object boundaries, thus it blurs the motion values around the object boundaries.

To overcome the constant smoothing across the whole chip, a second chip by Stocker (2002) implements a network which locally breaks the smoothness constraint at object boundaries. Motion segmentation is included in the computation by minimizing a modified cost function:

$$H_{OF}(v) = \sum_{ij} [(I_x v_x + I_y v_y + I_t)^2 + \rho_{ij}^x((\Delta^x v_x)^2) + (\Delta^x v_y)^2 + \rho_{ij}^y((\Delta^y v_x)^2) + (\Delta^y v_y)^2 + \sigma((v_x - v_{ref}^x)^2) + (v_y - v_{ref}^y)^2)] \quad (6.6)$$

where ρ_{ij}^x and ρ_{ij}^y are local variables that set the amount of smoothing. These variables are allowed to adapt locally or are set to zero in the case of an object boundary. The additional bias term in (6.6) allows components of the local optical flow vectors v_x and v_y to be compared against known motion priors v_{ref}^x and v_{ref}^y , respectively.

The advantage of the combination of the brightness constancy and spatial smoothness constraints lies in the robustness of the algorithm against noisy perturbations in the image. The smoothing operation across local motion outputs can average out this noise. The minimization of energy equations are attractive for aVLSI implementation because they do not require division circuits and no threshold operation is needed to avoid the

zero-division case. However, they do require high-precision multipliers with a wide linear input range as well as linear resistive networks with variable resistance that perform the smoothing operation. Since these circuits are difficult to implement in VLSI, both the implementations by Tanner and Mead (1986) and Stocker and Douglas (1999) showed a strong dependence of the motion output on stimulus contrast and can only produce reliable measurements with high-contrast stimuli.

Normal optical flow: Another way of estimating optical flow with gradient-based intensity algorithms is to compute normal flow. This computation is based on the assumption that of all possible optical flow vectors measured at a pixel, the correct flow is the one that is perpendicular to the edge orientation at each pixel. Circuits for implementing the normal flow model are proposed by Deutschmann and Koch (1998b); Mehta and Etienne-Cummings (2003) and Gruev and Etienne-Cummings (2004).

Mathematically, this constraint is set up as

$$\frac{I_x}{v_x} = \frac{I_y}{v_y}. \quad (6.7)$$

Substituting this constraint equation into (6.4) leads to:

$$v_x = -\frac{I_x I_t}{I_x^2 + I_y^2}, \quad v_y = -\frac{I_y I_t}{I_x^2 + I_y^2}. \quad (6.8)$$

The implementation of these equations in aVLSI circuits is challenging because the denominator in (6.8) includes partial derivatives of local brightness which are highly dependent on contrast. Especially for low contrast images one has to divide a small value in the numerator by another small number in the denominator. This division process is very susceptible to noisy estimates of the numerator and the denominator. Many aVLSI optical flow sensors include a threshold for I_x or I_y so that the optical flow is not computed when I_x or I_y are below this threshold. To avoid the zero-division problem, one implementation did not include the denominator term Deutschmann and Koch (1998b). Although this simplification reduces the complexity of the circuits, the optical flow output of the chip can be ambiguous and is highly dependent on contrast. Implementations of simplified forms of the normal optical flow equations $v_x = -\frac{I_t}{I_x}, v_y = -\frac{I_t}{I_y}$ decrease the complexity of the necessary circuits (Deutschmann and Koch, 1998b; Mehta and Etienne-Cummings, 2003). However, this simplification does not solve the zero-division problem. To avoid solving the zero-division problem on-chip, Gruev and Etienne-Cummings (2004) decided to compute only the local spatial and temporal derivatives $I_x, I_y,$ and I_t on-chip and to perform the division off-chip.

6.3.2 Intensity-based correlation

The intensity-based correlation algorithms are based on spatio-temporal frequency-based models of motion processing in flies as described by Reichardt (1961) and primates (Schiller et al., 1976). The primary spatio-temporal frequency-based motion algorithms that have been implemented in aVLSI are the **Hassenstein-Reichardt model** (Hassenstein and Reichardt, 1956), the **Barlow-Levick model** (Barlow and Levick, 1965) and the **Adelson-Bergen motion energy model** (Adelson and Bergen, 1985).

Many implementations of spatio-temporal frequency optical flow algorithms in single-chip aVLSI systems have been reported, for example, by Andreou et al. (1991), Gottardi and Yang (1993), Delbruck (1993), Meitzler et al. (1995), Harrison et al. (Harrison, 2005; Harrison and Koch, 1999), Liu (2000), and Higgins et al. (Higgins and Korrapati, 2000; Higgins et al., 2005; Pant and Higgins, 2007).

Intensity-based correlation algorithms work off the image brightness values unlike token-based algorithms that compute motion from detected features in the image. The motion outputs of the former algorithms have a profile that depends on the spatial and temporal frequency components of the local image patches in the field of view. Although the motion output amplitude is dependent on the square of the contrast of the input signals, these algorithms provide a motion output even for low-contrast inputs that fall below the threshold of a token-based algorithm.

The fact that the outputs of these models are dependent on both contrast and temporal frequency means that the readout is ambiguous. Thus it becomes impossible to determine the speed of a visual patch from a single filter. To solve this ambiguity problem, elaborated versions of these models use a bank of filters with different time constants (Santen and Sperling, 1984). These versions use a *place code* where the activation of a filter output codes for the presence of a particular spatio-temporal frequency component in the image patch instead of the value code used for gradient and time-of-travel algorithms. The place code helps for example, in situations where transparent objects are moving on top of one another; thus allowing several spatio-temporal frequency filters to be activated at the same time. A value code would have to decide for a particular movement or would just give an average response to the movements in the scene.

The place coding of optical flow using several filter banks is an advantage of spatio-temporal frequency-based motion algorithms but is also a drawback if implemented in aVLSI because of the following reasons: (a) Since several filters are needed for each motion pixel, the pixel fill-factor will be greatly reduced. (b) The readout and integration of the filter outputs is non-trivial even without considering mismatch. Mismatch between

the filter circuits in different pixels makes the readout very difficult because the time constants of the filters across pixels will not match.

6.3.3 Token-based correlation

Token-based correlation algorithms include an initial step of detecting a particular token or feature in the image. Common to all implementations is the use of a non-linear thresholding circuit for extracting local temporal contrast edges and/or spatial contrast edges. Contrast edges exceeding a threshold produce binary pulses. I only discuss two of the many correlation algorithms described in the literature: (a) The correlation is performed between a detected edge at one pixel with a delayed version of this edge at an adjacent pixel with the help of delay lines. (b) The correlation is performed between adjacent pixels using binary correlation filters that were inspired by the original Reichardt correlator.

Delay-line based correlation: Optical flow sensors that perform correlation based on delay lines (see the sensor by Horiuchi et al. (1991) for an example implementation) are inspired by the coincidence detector model in the auditory system of the barn owl. A temporal ON edge triggers a binary pulse of fixed width at each pixel. By propagating the pulses from neighboring pixels through two parallel delay lines from opposite directions optical flow can be determined from the intersection point of the pulses. The optical flow sensor by Horiuchi et al. (1991) highlights the following concerns: (a) aVLSI implementations of delay lines are generally costly in silicon area; (b) only a limited range of velocities can be detected for a particular delay setting.

Binary Reichardt correlator: The binary correlator chip by Sarpeshkar et al. (1993) is inspired by the Hassenstein-Reichardt correlator. Each edge that is detected by a optical flow pixel triggers a pulse of fixed width. This pulse is correlated with the delayed pulse of the neighboring pixel. The correlation circuit is very simple since the output is determined by the overlap of the two pulses. However, the pulse width determines the range of detected speeds, and the edge detection circuit by Sarpeshkar et al. (1993) is not sensitive to low-contrast stimuli.

Similar to the intensity-based correlation algorithms, the performance of the token-based implementations is limited by the fixed time constants of the correlation filters. To detect a wider range of stimulus velocities, filter banks with different time constants or correlation filters with self-adjustable delays would be necessary. These solutions would lead to even less compact designs and lower pixel fill factors. The clear advantage of token-based over intensity-based correlation algorithms lies in the non-linear edge de-

tection circuits that make them less dependent on contrast because each edge is simply represented by a binary event independent of contrast. However, where token-based correlation algorithms fail because the signal is below the threshold, intensity-based correlation implementations are often still able to report at least the correct direction of motion.

6.3.4 Time-of-travel

Time-of-travel algorithms directly measure the time taken by a contrast edge to travel between two adjacent pixels. Five major algorithms have been reported in the literature: facilitate-and-inhibit, facilitate-and-trigger, facilitate-trigger-and-inhibit, facilitate-and-sample, and facilitate-and compare. An overview of the different algorithms is given in figure 6.6.

One general advantage of time-of-travel motion detection chips is that they do not require any high precision dividers, multipliers and differentiators. Thus they do not suffer from zero-division problems like the gradient algorithms and allow for more compact designs than many correlation algorithms. The time-of-travel of a contrast edge is usually measured by using voltage or current pulses which are robust against noise perturbations. Due to the non-linear high-gain stages in the edge detection circuits, the motion outputs are dependent only on the stimulus speed and independent of the stimulus contrast down to very low contrast values.

All time-of-travel algorithms except the facilitate-and-sample algorithm that uses a log code (figure 6.6e), show a linear relationship between the time-of-travel t_{tof} of a contrast edge and their optical flow outputs. Thus the computed outputs are inversely proportional to the edge velocity v , that is, $v = \frac{\Delta x}{t_{tof}}$, where Δx corresponds to the inter-pixel distance. This inverse relationship means that the sensitivity of the algorithm to different velocity ranges is non-uniform. This property is common to all time-of-travel algorithms. As I show in chapter 7, it does not necessarily mean that these algorithms are at a disadvantage because the sensors can be tuned so that the sensitivity is approximately linear in the expected optical flow range.

Facilitate-and-inhibit: In the facilitate-and-inhibit (FI) algorithm (Figure 6.6a), an edge detected at a pixel triggers an output pulse which is reset when the edge reaches the neighboring pixel. The width of this pulse linearly increases with the time-of-travel of the stimulus.

The FI algorithm has its roots in the neural direction selectivity model of the rabbit retina. The first silicon design that implemented this model (Benson and Delbruck, 1992) uses

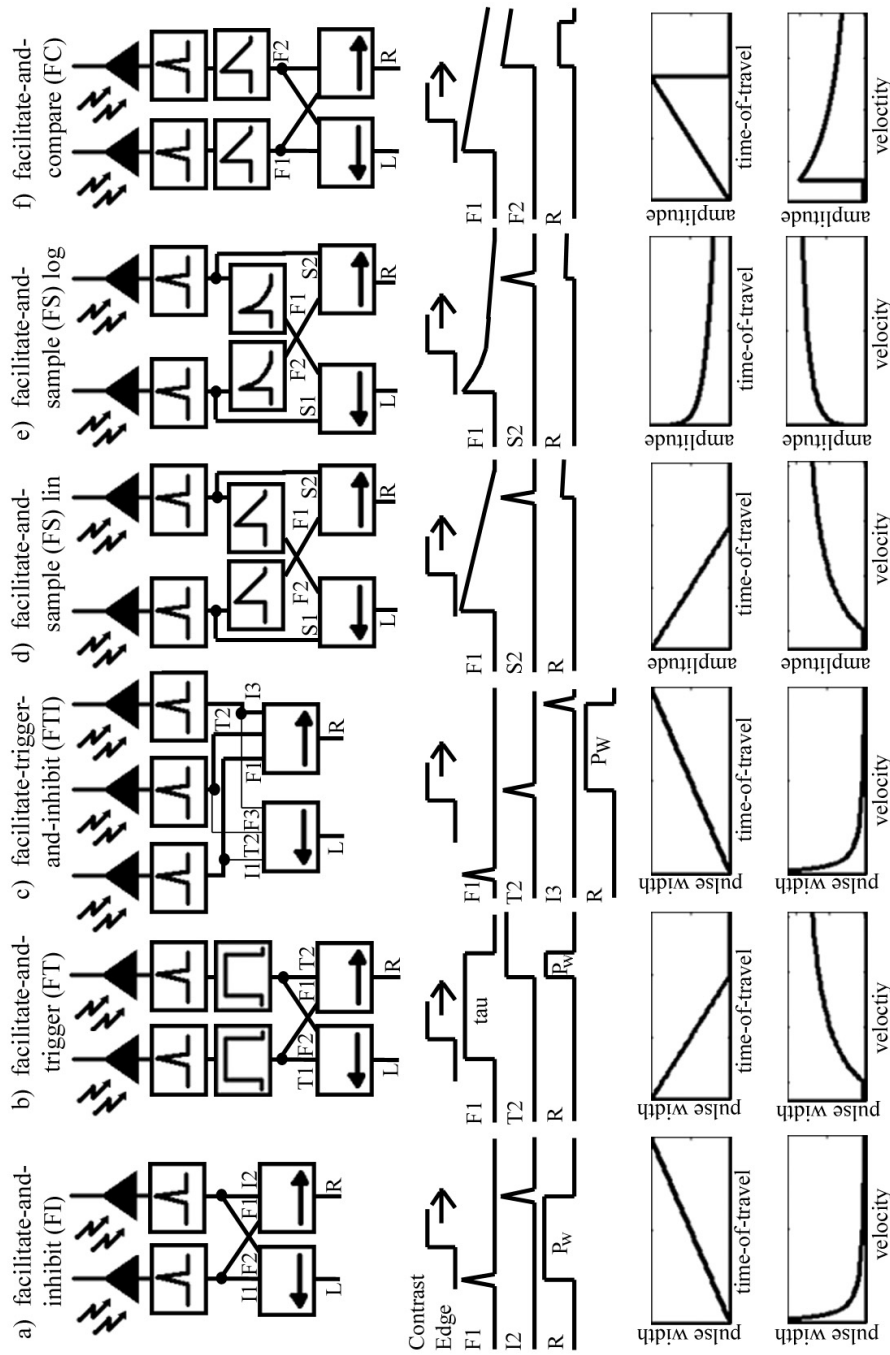


Figure 6.6: Comparison between time-of-travel motion algorithms. We show the block diagrams of circuitry (top), the time traces for the internal signals and the motion output that prefers a stimulus traveling from a photoreceptor at the left to its right neighbor (middle) as well as the dependence of the output signal on the time-of-travel and the velocity of the traveling stimulus (bottom). Note that while the FI, FT and the FTI algorithms use encode the stimulus velocity in the width of an output pulse the FS and FC algorithms output a value code. The labels note: F:facilitate, T:trigger, I:inhibit, L: optical flow output that prefers traveling edges from right to the left, R: optical flow output that prefers traveling edges from the left to the right, P_w :pulse width, tau:time constant of pulse with fixed width.

6.3. Evaluation of algorithms for optical flow sensors and their VLSI counterpart⁹³

temporal ON contrast edges to trigger pulses that are terminated by inhibition from neighboring photoreceptor pixels. In contrast to the motion chips that are described in the rest of this section, the implementation by Benson and Delbruck (1992) did not include a circuit that generates a binary pulse when an edge is detected. The output pulse amplitude of this circuit is dependent on the contrast of the temporal edge, thus making the chip usable only for limited contrast and velocity ranges.

An implementation that use pulses generated from the detection of a spatial edge using a spatial difference-of-Gaussian filter was proposed by Etienne-Cummings et al. (1993). The time-of-travel is determined by measuring the time that an edge took to appear at a pixel and disappear at the neighboring pixel.

Facilitate-and-trigger: In the facilitate-and-trigger (FT) algorithm (figure 6.6b) contrast edges trigger pulses of a fixed width. When an edge travels across the image plane it triggers two pulses at neighboring pixels that are delayed by the time-of-travel. Thus a simple boolean AND operation is sufficient to determine the temporal overlap of the pulses which linearly decreases with the image velocity. The FT implementation by Kramer et al. (1997) can detect image velocities from 2 mm/s to 120 mm/s, measured through image projections onto the chip. However, as shown in figure 6.6b the motion output saturates for higher image velocities. The general problem of the FT algorithm lies in the fixed pulse width of the triggered pulses. If the pulse width is too small, slow stimuli cannot be detected since the pulses will not overlap. On the other hand, a large pulse width can lead to an output that constantly remains 'high', for example when pulses are triggered periodically due to a highly textured and fast stimulus. This means that no output pulse is detected by the system.

Facilitate-trigger-and-inhibit: To overcome the fixed pulse width limitation in the FT algorithm, Kramer (1996) introduced the facilitate-trigger-and-inhibit (FTI). The FTI algorithm (Figure 6.6c) integrates signals from three adjacent pixels and outputs a binary pulse whose width is inversely related to the velocity of contrast edges traveling across the image plane. A contrast edge moving from left to right causes the edge detection circuit at pixel 1 to output a pulse, called the facilitate signal F1. The signal F1 enables the motion detection circuit to generate the trigger signal T2, that is caused by the same edge traveling across pixel 2. The signal T2 causes the motion circuit to output a pulse R that lasts until the edge reaches pixel 3. Once the edge is detected by pixel 3, it sends an inhibition signal I3 which terminates the output pulse R. The detectable image speed reported in Kramer (1996) ranges between 0.034 mm/s to 60 mm/s. For velocities above 60 mm/s the circuit also shows a motion output in the non-preferred direction due to the limited rise time of the inhibition signal.

Facilitate-and-sample: Facilitate-and-sample (FS) aVLSI optical flow sensors were first described by Kramer et al. (1995, 1997). FS sensors use a contrast edge detected by a pixel to trigger both a small sampling pulse (S1, S2) as well as a facilitate signal (F1, F2) that linearly or logarithmically decays over time (figure 6.6d,e). The time to travel across two pixels for a contrast edge is determined by the sampled value of F1 by S2 for stimuli moving from left to right and by the sampled value of F2 by S1 for stimuli moving in the opposite direction. A similar token-based algorithm implemented with discrete electronics components was reported earlier by Franceschini et al. (Blanes, 1986; Franceschini et al., 1986). The work by Kramer and Franceschini had great influence on the design of the dynamic optical flow sensor which I later describe in (chapter 7).

The FS optical flow sensor of Kramer uses a logarithmic encoding for the time-of-travel (figure 6.6e). This encoding allows the motion circuit to represent a time-of-travel range of more than seven orders of magnitude for fixed time constants. This chip performance is the best reported in the literature so far. However, these motion measurements were done by using electronically generated pulses which bypassed the outputs of the pixels' edge-detection circuits. In addition, due to the highly compressed encoding of the motion output, where a change of 200mV in the output corresponds to one order of change in the time-of-travel of an edge, 20mV of noise will lead to a relative error of 10% in the decoding of the time-of-travel. So the optical flow output is very vulnerable to noise.

The FS algorithm is advantageous over the FI, FT, and FTI algorithms because its motion output can be held on a capacitor for a few ms, while the latter algorithms produce temporal-coded motion outputs that need to be constantly monitored.

Facilitate-and-compare: The facilitate-and-compare (FC) algorithm is inspired by the FS algorithm (Deutschmann and Koch, 1998a). The implemented algorithm (figure 6.6f) does not require sample-and-hold circuits to store the sampled value of the facilitate signals. The optical flow output is computed continuously by subtracting the linearly-decaying facilitate signals from adjacent pixels. This difference codes the speed of the stimulus. This algorithm leads to less complexity for the FC circuits but the implemented circuits suffer from the tradeoff between high sensitivity settings (the facilitate signals will decay quickly to ground) and from mismatch (the motion output varies over time if the decay rates of the facilitate signals are different). The FC implementation by Deutschmann and Koch (1998a) can measure image velocities in the range from 0.3 mm/s to 40 mm/s.

6.4 Conclusions

Over the last 2 decades, more than 50 different aVLSI monolithic motion detection chips have been reported. The implemented algorithms on these chips can be divided into more than 14 sub-categories. Because of the wide range of algorithms and implementations, an unbiased comparison of the best optical flow sensor to use on a robotic platform is difficult. There are several things that one can consider when designing these sensors for a mobile robot:

- To operate in natural environments and without needing high-contrast textures, the chip should respond to low contrast stimuli with small spatial frequencies reliably and unambiguously.
- The chip should work robustly under different background light intensities.
- The optical flow sensor outputs should have low mismatch or the mismatch must at least be easily removed in the readout, for example, through calibration.
- The chip should operate reliably in the presence of noise for example, noise on the power supply lines as well as noise in the visual stimuli like light flicker.
- The pixel design should be compact and the design should support a large two-dimensional spatial array.
- Since power resources on autonomous flying platforms are limited, optical flow sensors should consume μW of power.

To decide on a particular optical flow algorithm and its subsequent implementation, one has to clearly identify the advantages and disadvantages that come from the motion algorithm itself and those that come from the VLSI circuit implementations. For example most of the motion detection chips reported so far show a high dependence of the optical flow outputs on the stimulus contrast. While this **contrast sensitivity** can be easily explained by the algorithm itself for intensity-based correlation implementations, there is no theoretical reason why a FT implementation should respond more ambiguously to low contrast features than an FS implementation. In this case, the ambiguous response of the FT motion chip lies in the implementation of the edge detection circuits.

There is no single optical flow sensor in the literature so far that fulfills all my evaluation criteria. Since most of these sensors have been designed to demonstrate a particular principle or a specific novel circuit, the sensors are often not optimal for operation on

a MAV. Coupled with this is the fact that there are no chip specifications for the end-user. This is why researchers that want to explore new control algorithms end up using cameras on their platform.

In my opinion the most interesting optical flow sensors presented so far are the normal flow implementations by Mehta and Etienne-Cummings (2006), the gradient-based implementation by Stocker (2002), the edge-counting sensor by Arreguit et al. (1996) and the facilitate-and-sample implementation by Kramer et al. (1995, 1997).

- **The normal flow implementations by Mehta and Etienne-Cummings (2006)**

use a clocked active pixel sensor (APS) front-end as well as correlated double sampling (CDS) for on-chip mismatch compensation. The clocked-based APS circuits allow for a sequential readout of local intensity values of a row of pixels over a global data bus. The computation of the optical flow values is done on the periphery of the APS array so very dense 2-D pixel arrays can be achieved. The authors report of an array size of 120 by 80 pixels, a fill-factor of 10%, and an equivalent frame-rate of 20 frames per sec with a power consumption of 34.5mW. Note that the normal flow estimation in this implementation is done in a line-sequential way. The mismatch between the outputs of the different APS pixels is removed by the correlated double sampling (CDS) technique and since the motion values are computed sequentially row by row, the circuits that perform the motion computation can be shared thus eliminating another source of mismatch.

However, this implementation has drawbacks that are caused in part by the APS implementation and by the normal flow algorithm itself. Since the operation of the APS requires a clock, this implementation is susceptible to temporal aliasing unlike the continuous-time optical flow sensor chips. In addition, the sequential readout of the APS array for computing the motion can be a bottleneck and prevents the detection of fast moving stimuli.

A general problem with the normal optical flow implementation is the need for high precision division circuits whose output is imprecise for low contrast stimuli and especially in the presence of mismatch and noise. These chips cannot produce reliable measurements in low contrast environments and a thresholding circuit is needed to eliminate low-contrast signals from the flow computation. The implementation by Mehta and Etienne-Cummings (2006) does not include any circuits for spatial integration or segmentation.

- **The gradient-based implementation by Stocker (2002)** is an improved version of the sensor by Stocker and Douglas (1999). It is the only optical flow sensor that

I know of that implements the circuits of optical flow computation together with circuits for spatial integration addressing the aperture problem as well as circuits for motion segmentation on a single chip. The sensor operates on continuous time and uses the adaptive photoreceptor circuit with selectable adaptation time constants by Liu (1999) which allows circuit operation over a wide range of background lighting conditions. The need for linear multiplication circuits makes the chip vulnerable to noise and mismatch.

Due to the strong dependence of the optical flow outputs on stimulus contrast for contrast values below 50%, the chip is not usable in low-contrast environments. The smoothing and multiplication operation circuits also require a lot of transistors leading to a reported pixel fill factor of 4%. Furthermore, although the chip performs smoothing of optical flow values, the local flow values need to be read out from all pixels individually. There is no optical flow output that is the result of a spatial integration of several pixels. Finally, I do not believe that the spatial integration is optimally implemented. The weight that sets how strong neighboring pixels are coupled is set to be constant except at neighboring pixels that provide a sufficiently different optical flow output to trigger the segmentation circuits. Also the segmentation threshold is fixed. Both constant coupling weight and constant segmentation threshold can lead to incorrect optical flow outputs since their correct values typically depends on the particular scene that is projected onto the sensor.

- **Arreguit et al. (1996)** described a computer mouse chip that computes optical flow by **counting the number of edges traveling across the focal plane** within a certain time bin. The chip contains 75 pixels arranged in the form of a circular spot. It was developed to replace the mechanical parts of the track ball in a computer mouse and was designed to measure the rotation of the track ball. The authors showed that the displacement of the ball in the x - and y -direction Δx and Δy could be determined using the following equations:

$$\frac{\Delta x}{P} = \frac{\sum R - \sum L}{\sum E_x}, \quad (6.9)$$

$$\frac{\Delta y}{P} = \frac{\sum U - \sum D}{\sum E_y} \quad (6.10)$$

where P represents the pitch between the pixels, and $\sum R$, $\sum L$, $\sum U$, and $\sum D$ represent the number of edges that traveled to the right, left, upwards or downwards, respectively, within the time bin. $\sum E_x$, and $\sum E_y$ represent the total number of vertical and horizontal edges, respectively. An interesting property of the

chip was that its resolution increased with the number of edges and the minimum detectable displacement was smaller than the distance between the photodiodes P . The authors reported a resolution of more than 800 dpi over a 100-2000 mW/m^2 light intensity range.

The continuous-time front-end of the optical flow sensor was highly optimized to work together with a pulsed LED and a tracking ball with high contrast patterns. With the presented front-end the sensor is thus not usable for controlling MAVs but it introduces interesting ideas and circuits: The design by Arreguit et al. (1996) is an example how spatial integration across an optical flow sensor can be implemented in an elegant way where the entire sensor directly outputs a global optical flow signal. The circuits for counting edges are simple and can be implemented in a compact way. Furthermore, the total amount of edges that have contributed to the optical flow calculation per unit time can form a well-suited reliability signal.

- **Facilitate-and-sample implementations like the one presented by Kramer et al. (1995, 1997)** look interesting to me because they can be implemented using continuous time circuits and are less vulnerable to circuit noise and mismatch. Facilitate-and-sample implementations can be made to consume little power if one uses the fact that optical flow only needs to be calculated when a feature is detected. Except for the photoreceptor front-end, the circuits of pixels where no feature is present in theory could be deactivated and quickly reactivated when being needed again.

The sensor front-end is usually implemented using the adaptive photoreceptor circuit by Delbrück and Mead (1994). This circuit has been shown to be able to adapt locally at each pixel over 6 decades of background light intensity thus allowing a reliable coding of the motion outputs under conditions ranging from moonlight to sunlight. This adaptation property is of great benefit when the chip is operated in natural surroundings where there could be a wide spatial distribution of light intensities, for example, when sunlight streams into part of a room and the remaining areas are in shadow.

The drawback of the facilitate-and-sample implementations is that optical flow is not calculated continuously but only if a contrast edge is detected. Thus the overall sensor performance greatly depends on the edge detection circuit. The edge detection circuit used by Kramer et al. (1995, 1997) for example makes the optical flow outputs depending on contrast.

Based on the evaluation of optical flow sensor implementations in this chapter, I concentrated on two main ways of extracting flow in this thesis:

1. The design of my monolithic dynamic optical flow sensor (DOFS) presented in chapter 7. Like the optical flow sensor presented by Kramer et al. (1995, 1997), DOFS implements a facilitate-and-sample algorithm to extract optical flow and uses the adaptive photoreceptor circuit by Delbrück and Mead (1994). However, for this sensor I developed a novel high-gain feature detector that improves the sensitivity of DOFS to low contrast features by a factor greater than 7 in comparison to the facilitate-and-sample sensor by Kramer et al..
2. The development of an optical flow sensor module for ultra-lightweight flying robots presented in chapter 9. The sensor module is composed of my adaptive vision sensor (aVIS) that implements the continuous-time adaptive photoreceptor front-end by Liu (1999) and a microcontroller running the Image Interpolation Algorithm proposed by Srinivasan (1994) for extracting optical flow from the outputs of aVIS.

Chapter 7

Dynamic optical flow sensor

Abstract

This chapter presents a novel dynamic optical flow sensor (DOFS) that extracts optical flow information by measuring the time a visual contrast edge requires to travel between two adjacent photoreceptors on the sensor's focal plane. This sensor implements the facilitate-and-sample (FS) algorithm. It includes an adaptive logarithmic wide-dynamic-range photoreceptor front-end by Delbrück and Mead (1994, 1995); a modified LMC circuit which removes DC and amplifies transient changes, and a novel double-threshold edge-detection circuit which extracts edges from even low-contrast stimuli.

Section 7.1 gives an overview over the DOFS and motivates design choices while section 7.2 shows the implemented circuits in detail. Section 7.3 presents experimental results from the fabricated sensor. Section 7.4 discusses results and concludes the chapter.

Highlights and original contributions

This section presents the following highlights and original contributions:

1. A novel **double threshold circuit** that allows for a robust edge detection for low contrast stimuli in the presence of noise. Using two separate threshold parameters the circuit allows to dynamically trade between the minimum contrast a visual stimuli must have to be detected and the amount of input signal noise that is rejected during the edge detection process. This circuit has major impact on

reducing the dependence of the sensor's optical flow outputs on the contrast of the visual stimuli and its robustness to noise (section 7.2.3).

2. A **modified LMC circuit** that amplifies temporal contrast of visual stimuli and processes the outputs of the adaptive photoreceptor circuit by Delbrück and Mead (1994, 1995) to provide an optimized output signal for my edge detection circuit (section 7.2.2).
3. I propose circuits for **FS optical flow sensors with outputs that are linearly proportional to stimulus velocity** - not time-of-travel. (section 7.4)

7.1 Dynamic optical flow sensor - overview, motivations and properties

This section introduces the dynamic optical flow sensor (DOFS) and motivates the implementation of a facilitate-and-sample (FS) sensor in analog VLSI technology.

7.1.1 Sensor overview

Table 7.1: Specification of dynamic optical flow sensor and lens.

Fabrication process	2-metal 2-poly 1.5 μm CMOS process
Supply voltage	5.0V
Number of pixels	24 x 1
Pixel distance	54.4 μm
Pixel size	54.4 μm x 1521.6 μm
Size of photodiode	44 μm x 44 μm
Chip size	2.2mm x 2.2mm
Adaptation to background light over	4 decades
Lens	Computar 1/3"
Focal length	8.0mm
F/#	1:1.2

Table 7.1 gives an overview of the properties of the dynamic optical flow sensor (DOFS). Pictures of the packaged sensor are shown in figure 7.1. DOFS contains six main

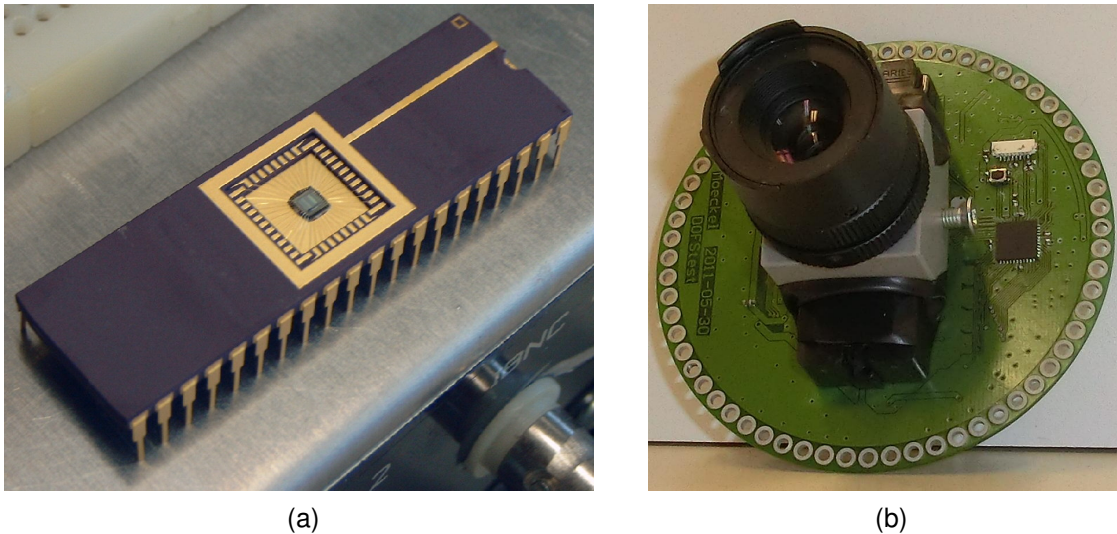


Figure 7.1: Pictures of (a) DOFS integrated into a DIP package with 40 pins without lens and (b) with lens mounted on printed circuits board.

circuit blocks that have to work together to estimate optical flow from visual contrast edges traveling across the sensor's focal plane. Figure 7.2 gives an overview of these circuits. The figure shows three neighboring pixels that form a linear array. Like most token-based visual sensors, DOFS contains elements for phototransduction that convert light falling onto the sensor's focal plane into electrical signals (depicted as photodiode symbols DL, D, and DR in figure 7.2), analog circuits that preprocess and filter these voltage or current signals (PR, LMC), circuits that detect tokens (edge detection block) and finally circuits that implement the optical flow estimation by sharing information between neighboring pixels (pulse shaping, facilitate signal, sample & hold & decay circuit). DOFS also contains a **scanner** circuit (section 7.2.6) that allows the sequential readout of local analog pixel values (Mead and Delbruck, 1991).

In the DOFS facilitate-and-sample (FS) implementation, photocurrent generated in photodiodes is converted into a voltage signal by an **adaptive photoreceptor (PR)** circuit (section 7.2.1) that adapts over four decades of background illumination. An analog **LMC** circuit (section 7.2.2) outputs a high gain, high-pass filtered version of the photoreceptor signal, V_{lmc} , that is processed by a novel double-threshold **edge detection** circuit (section 7.2.3). This edge detection circuit generates a digital event, V_{dt} , in the presence of a temporal contrast edges in the LMC output signal.

The FS algorithm is implemented by the following circuits: The **pulse shaping** circuit (section 7.2.4) generates an asynchronous short digital pulse, P , every time a contrast edge is found by the edge detection circuit. Each contrast edge triggers the reset of

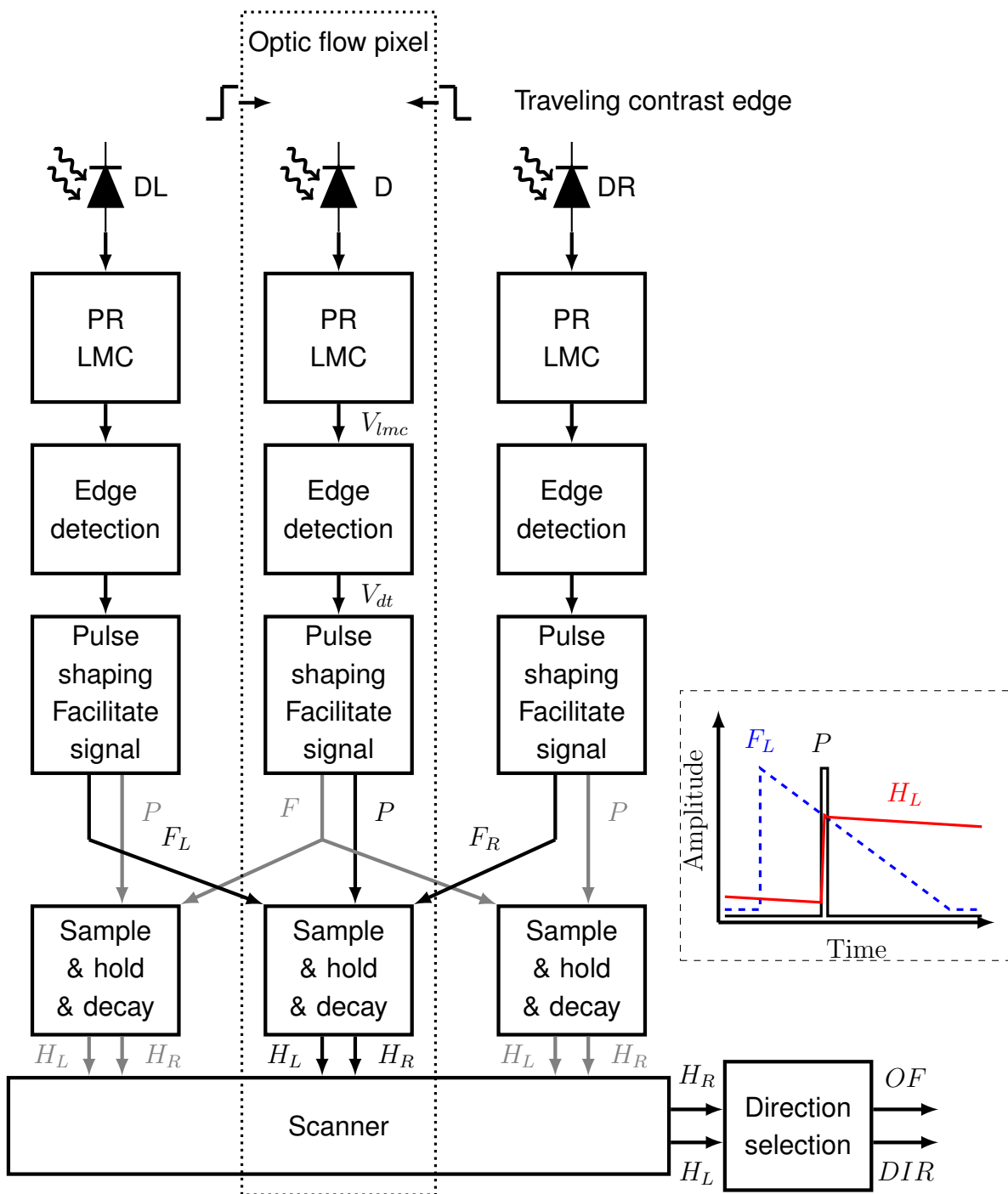


Figure 7.2: Overview of dynamic optical flow sensor circuits.

a **facilitate signal**, F , (described in section 7.2.4) that decays over time as depicted by the blue dashed line in the timing diagram of figure 7.2. The facilitate signal of neighboring left (F_L) and right (F_R) pixels are sampled at the time when the pulse P

is generated by the **sample & hold & decay** circuit (section 7.2.5). If for example a contrast edge travels from the left to the middle pixel it first triggers the reset of the facilitate signal F_L that decays over time while the contrast edge travels to the middle pixel. Once the contrast edge reaches the middle pixel it triggers the generation of the sampling pulse P that is used to sample F_L and to update H_L which stores the sampled value as shown in the timing diagram of figure 7.2. The amount of decay of F_L at the time when P is generated reflects the time-of-travel of the contrast edge. The greater the value of F_L , the faster the contrast edge was traveling, the higher the optical flow as well as the value H_L . A contrast edge traveling from the right to the middle pixel would trigger the reset of F_R first that would be sampled when the edge reaches the middle pixel and stored as H_R . So by comparing the absolute values H_L and H_R one can find the direction of image motion. The storage devices that hold H_L and H_R can be digital memories when the facilitate signals are implemented in form of digital counters. In this design, I decided for analog facilitate signals that I sample onto analog capacitors where the signals slowly decay over time.

The **direction selection** circuit (section 7.2.7) compares the absolute values of H_L and H_R to estimate the direction. If a contrast edge travels from the left to the right pixel the reset of F_L precedes P and the reset of F_R . The value sampled onto H_L is thus higher than H_R and the direction selection circuit sets the digital direction flag DIR accordingly and selects the optical flow output OF to be equal to H_L . If the contrast edge travels from the right to the left side, F_R gets reset first, H_R is higher than H_L and the optical flow output OF is set equal to H_R while the direction flag DIR is inverted.

In this design, the H_L and H_R signals are not optimally chosen for comparison by the direction selection circuit. In contrast to Kramer et al. (1995, 1997) I compare the time-of-travel between three instead of two neighboring pixels' photodiodes. Thus if for example two contrast edges travel in opposite directions as shown in figure 7.2 the optical flow outputs can be corrupted. This is why for the experiments shown in this chapter, I typically ignore the outputs of the direction selection circuit and directly read the local sample-and-hold signals H_L and H_R . I then compare the H_L and H_R values of neighboring pixels in software, remove offsets and correct for the selection problem.

Figures 7.3a and 7.3b show the layout and a microphotograph of the fabricated sensor chip. DOFS comprises 24 optical flow pixels implemented on a 2.2mm x 2.2mm chip area in a 2-poly 1.5 μ CMOS process. As discussed in section 6.1 to obtain a good spatial resolution, I decided to layout the pixels in form of a 1-dimensional linear array. This design decision was taken because the sensor was originally intended for use on the 10-gram airplane by Zufferey et al. (2007). As presented in section 4.1 Zufferey

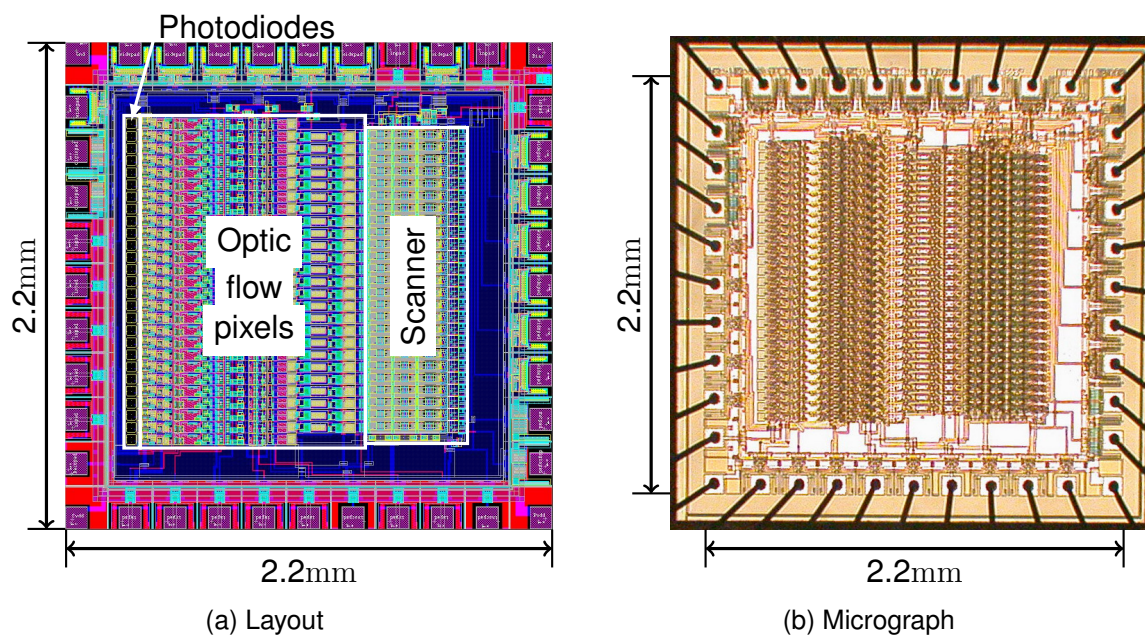


Figure 7.3: (a) Layout and (b) microphotograph of dynamic optical flow sensor. The microphotograph was taken by Delbrück.

et al. have demonstrated that linear cameras are sufficient for autonomous optical flow based control their airplane.

The circuits and the optical flow algorithm implemented in this chip are inspired by biology. The photoreceptor front-end mimics the background light adaptation of the visual system of many animals (Delbrück and Mead, 1994). The output of the photoreceptor is processed by the LMC circuit that models the function of the Laminar Monopolar Cell in the fly visual system (Liu, 2000). The time-of-travel optical flow algorithm was inspired by the fly visual system (Franceschini et al., 1986).

7.1.2 Properties of the facilitate-and-sample circuit implementation

Design choices for a robust optical flow sensor implementation

I chose to implement the facilitate-and-sample (FS) algorithm because according to the evaluation of existing sensors and optical flow algorithms (section 6), the FS method is one of the best suited algorithms for implementation in analog VLSI technology and for the visual guidance of autonomous robots (section 6.2):

- The general structure of the token-based algorithm where the token detection circuit separates the analog circuits at the sensor's front-end from the circuits that process the token event allows for robust VLSI implementations. While the analog circuits at the front-end are still relatively vulnerable to noise in the visual stimuli and circuit mismatch the token-processing circuits mostly need to deal with **asynchronous digital signals** that **are robust to noise**.
- Since the facilitate-and-sample (FS) algorithm (in contrast to many other optical flow algorithms) **does not require any complex or high precision numerical operations** like multiplications, divisions, additions or subtractions, but is based on relatively simple reset, decay, sample and pulse-shaping operations, it is less sensitive to circuit noise and mismatch. The critical part of the FS implementation is the reliable detection of contrast edges and the generation of a pulse whose properties do not vary for different types of visual stimuli.
- The FS algorithm allows for a **continuous time** implementation with analog circuits and without an external clock signal for synchronization. All pixels perform **parallel low-latency optical flow estimations** thus supporting the utilization of the sensor in low-latency control applications. The bottleneck in the present DOFS design is the scanner circuit that needs a clock for sequential access to local pixel values.
- The FS algorithm supports the utilization of the adaptive photoreceptor (PR) front-end that makes the design **robust to variations of background light intensity** (Delbrück and Mead, 1994, 1995).
- Both the PR and LMC circuits amplify temporal contrast edges of small amplitude that allows the DOFS to operate on **low contrast visual stimuli**.
- I use several mechanisms to **reduce the effect of circuit noise**:
 - I use the biases of the PR and LMC circuits as well as the one of their intermediate buffer circuits (see section 7.2.1) to control the circuits' bandwidth and filter both noise generated by the circuits (discussed in section 5.3) as well as noise caused by the visual stimuli (like flicker noise from artificial light sources).
 - The double-threshold edge detection circuit (section 7.2.3) allows for filtering further noise of higher amplitudes and lower frequencies that cannot be removed by the PR and LMC circuits. Its high gain amplifiers ensure that the

shape of the edge detection signal, V_{dt} , is independent from the contrast of the visual stimuli.

- The pulse-shaping circuit generates a pulse with a refractory period in which no additional pulse can be generated. This ensures that contrast edges cannot be accidentally detected within a short period of time.
- In contrast to Kramer et al. (1997), I do not use a logarithmic but a linear encoding for the time-of-travel information. Although the logarithmic encoding allows the sensor by Kramer et al. (1997) to operate over stimuli velocities of seven orders of magnitude for fixed time constants, the highly compressed encoding of the optical flow output make it very vulnerable to the addition of noise. 20mV of noise will lead to a relative error of 10% in the decoding of the time-of-travel. Furthermore, Kramer et al. (1997) could only show motion measurements over a large range of stimulus velocities by using electronically generated pulses which bypassed the outputs of the pixels' edge-detection circuits. For my DOFS, I decided for a linear encoding of the time-of-travel which requires a constant update of the facilitate signal's time constant to operate over more than two orders of magnitude but which makes the sensor's outputs less vulnerable to noise.

Expected properties of the FS sensor's optical flow outputs

Due to the FS algorithm, the dynamic optical flow sensor inherits the following properties:

- FS sensors **measure the time-of-travel**, t , of contrast edges. It is common that the facilitate signal, F , is reset to its maximum value, F_{max} , and decays over time so that the sensor outputs a signal that increases with stimulus velocity. However, for a linear decay of the facilitate signal, the sensor's **optical flow output**, OF , is **not linear with respect to the velocity of the visual stimuli**, v , but follows the relationship

$$\begin{aligned} OF &= OF_{max} - \frac{\Delta F}{\Delta t} \cdot t \\ &= OF_{max} - \frac{\Delta F}{\Delta t} \cdot \frac{s}{v} \end{aligned} \quad (7.1)$$

where OF_{max} corresponds to the maximum value of OF (typically equal to F_{max} except for some offsets due to circuit limitations), s is the distance between the

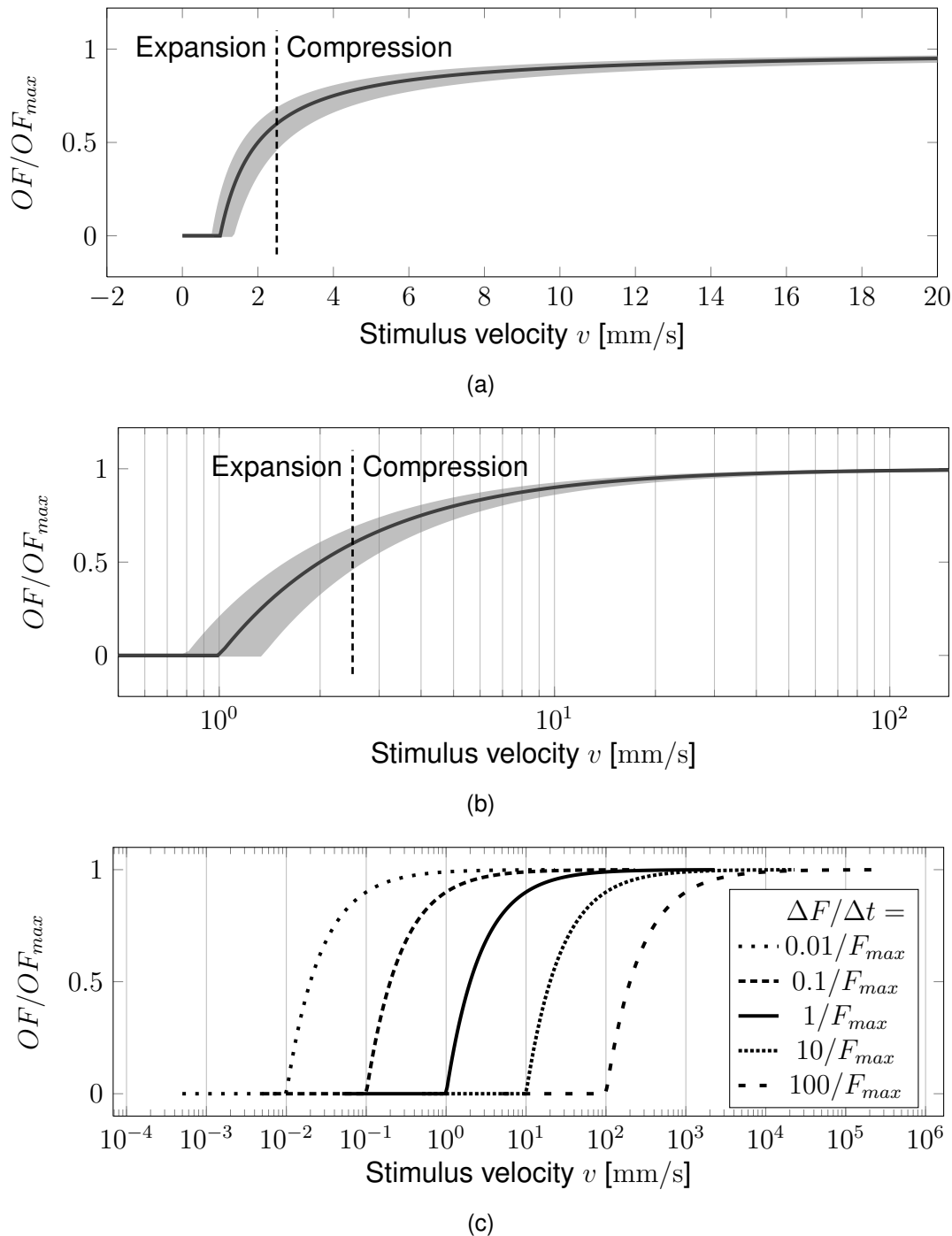


Figure 7.4: (a) Linear and (b) logarithmic plot of the optical flow output signal generated by the facilitate-and-sample algorithm with respect to stimulus velocity. Solid lines depict the sensors optical flow output OF normalized to the maximum facilitate signal F_{max} . Colored areas show the variance of the OF -signal in response to variations of the stimulus velocity with a standard deviation of 50%. The OF outputs were calculated for a decay rate $\Delta F/\Delta t = 1/F_{max}$ of 1/s and a pixel distance of $s = 1mm$. (c) By adjusting the decay rate a single FS sensor can respond to stimulus velocities that vary over several orders of magnitude.

sensor's pixels, v is the velocity of the visual stimulus, and $\Delta F/\Delta t$ is the decay rate of the F -signal. Note that OF cannot become negative.

Figure 7.4 shows the theoretically expected dependence of the FS sensor's optical flow outputs on the velocity of the visual stimuli traveling across the sensor's focal plane. The non-linear relationship between the OF output and stimulus velocity v can be a drawback for controllers that rely on a linear dependence of both signals. However, if properly used the non-linearity becomes an advantage: When constantly adapting the decay rate $\Delta F/\Delta t$ the FS sensor can cover several orders of magnitude of stimulus velocities (figure 7.4c) where for each decay rate there is a region in the OF - v relationship where even small variations of the stimulus velocity can be resolved (expansion region in figures 7.4a and 7.4b) and a region where the OF -signal saturates (compression region in figures 7.4a and 7.4b). During operation it is advantageous to continuously adjust the decay rate $\Delta F/\Delta t$ so that the expected stimulus velocities fall into the expansion region and can be measured with optimal precision. The compression region is useful since it allows the estimation of stimulus velocities that do not fall into the expansion region and a fast adaptation of the decay rate $\Delta F/\Delta t$.

The FS algorithm is versatile and allows also for other OF - v output profiles. As will be discussed in section 7.4, a linear or more complex OF - v relationship can be achieved through the application of several decay rates and threshold operations. For example, Kramer et al. (1995, 1997) obtained a logarithmic decay rate through a diode-connected transistor.

- In contrast to other optical flow algorithms (for example gradient-based algorithms), the FS algorithm does not provide a continuous update of optical flow output values but only in the event of a detected contrast edge. The FS algorithm is advantageous over other time-of-travel algorithms (see chapter 6 for details) since the optical flow output is stored on a storage device and does not need to be constantly monitored. In any case the event-based optical flow computation can be a challenge if a controller requires the continuous update of each pixel but also opens the potential for event-based communication and optical flow processing.

7.1.3 Sensor history, modifications and acknowledgment

I based my DOFS design on an unpublished implementation of the facilitate-and-sample algorithm by Jörg Kramer and my supervisor Shih-Chii Liu from the Institute of Neuroin-

formatics. Based on this earlier design, I made additional modifications:

1. I designed a new edge detection circuit since I found that both the original designs by Kramer et al. (1995, 1997) as well as in the unpublished sensor, suffer from a strong dependence of the optical flow outputs on the contrast of the visual stimuli. From simulations and by testing the unpublished hardware sensor, I identified the edge detection circuits as the primary cause for this contrast dependence.
2. I designed my edge detection circuit using two global thresholds which allows for a reliable feature detection in the presence of noise.
3. I modified the LMC circuit that amplifies and rectifies the signals from the photoreceptor circuit by adding a circuit by Liu that allows for adjustment of the adaptation time constant of the LMC circuit. Furthermore, in contrast to Kramer et al. (1995, 1997) I decided to use the voltage output of the circuit instead of the current output as the input to my edge detection circuit.
4. I replaced the logarithmic encoder by Kramer et al. (1995) with a circuit that uses a linear encoding for the time-of-travel since this makes the optical flow outputs less vulnerable to the addition of noise. By updating the time constant of the linear encoder in a feedback loop, one can still measure optical flow values that vary over several decades of amplitude (see section 7.3).
5. In contrast to the unpublished sensor which uses local direction selective circuits for each pixel, I use a global direction selection circuit placed at the output of the on-chip scanner circuit. This way the required chip area as well as the effect of fabrication mismatch are reduced since all local optical flow signals are processed by the very same circuit.

7.1.4 Remarks on circuit layout and fabrication process

The fabrication of chip prototypes is expensive. The first DOFS was fabricated in a relatively low-cost but also relatively outdated 2-metal 2-poly $1.5\mu\text{m}$ CMOS process. The restriction of two metal layers and the relatively large minimal transistor size of $1.5\mu\text{m}$ lead to a relatively large sensor pixel layout. Modern chip fabrication processes for analog circuits feature minimum sizes of 180nm and several metal interconnect layers allowing for a more dense implementation of complex circuits. The fabrication process

however supports the design of relatively well matched poly1-poly2 capacitors with relatively high capacitance per area. These capacitors are used to reduce the mismatch between different optical flow sensor pixels.

The FS implementation was optimized for providing robust optical flow outputs - not for overall pixel size. Each FS pixel in this design requires 7 analog capacitors as well as 70 MOSFETs from which 58 are analog MOSFETs that are not implemented with minimum size to reduce the effect of fabrication mismatch. The scanner circuit requires 21 digital MOSFETs per pixel as well as 5 analog and one digital MOSFET per pixel. The overall required pixel area when implemented in the 2-metal 2-poly $1.5\mu\text{m}$ CMOS process is $54.4\mu\text{m} \times 1521.6\mu\text{m}$ including the scanner for reading 6 local pixel values and $54.4\mu\text{m} \times 1063.2\mu\text{m}$ excluding the scanner circuit.

Since visual sensors like DOFS are constantly exposed to light, analog circuits need to be shielded from undesired photo-generated current. This is why all the analog circuits except for the photodiodes in the DOFS are covered with a metal shield that is also connected to the power ground of the circuits. For connecting power VDD, interconnection of circuits and for setting circuit parameters the remaining metal layer and the poly1 layer have been used. Care was taken that wires that carry higher currents like the power supply or that connect to the source or drain of MOSFETs are typically made from metal. Poly1-wires that have a higher resistance than metal wires are only used to drive the gates of MOSFETs and carry those signals that do not require high currents.

Under all these constraints the limited chip area of $2.2\text{mm} \times 2.2\text{mm}$ allowed the placement of 24 optical flow pixels and 40 pads. The limited amount of available pads that form the input and output terminals of the sensor required sharing several parameters among the circuits and making trade-offs, This is why not to all circuits optimal analog parameter values can be applied.

7.2 Circuits of the dynamic optical flow sensor

This section discusses in detail the individual circuits of the dynamic optical flow sensor.

7.2.1 Adaptive logarithmic wide-dynamic-range photoreceptor circuit

The inclusion of circuits for **background light adaptation at pixel level** allows each pixel to adapt individually. This is in contrast to a visual sensor array in a standard camera that globally controls the exposure time of all pixels. If the entire sensor experiences the same amount of background light a global adaptation mechanism is sufficient and can lead to lower fabrication mismatch. However, when different parts of the sensor are exposed to different background light intensities because for example the visual scene is composed of bright sunlight spots as well as dark shadows global adaptation mechanisms tend to fail and the local adaptation of individual pixels becomes an advantage.

DOFS uses the adaptive logarithmic wide-dynamic-range photoreceptor first described by Delbrück and Mead (1994, 1995). This photoreceptor (PR) is an evolution of the first adaptive photoreceptor design by the same authors (Delbrueck and Mead, 1989). It amplifies temporal contrast and adapts locally to changes of background light over several orders of magnitude.

Transfer function of the photoreceptor circuit

Over a large range of light intensities, the photocurrent generated by a photodiode is directly proportional to the intensity of the light falling onto the photodiode's active area. As a result the generated photocurrent varies over several orders of magnitude when a visual sensor should be operated both under bright sun light as well as indoors in rooms with artificial light sources, in shadows, or even during the night under moon light. Under this variety of background light conditions the photocurrent of a photodiode with an area of about $10\mu\text{m}^2$ varies between a few pA and up to several hundreds of nA.

In a closed-loop combination as shown in figure 7.5c, the inverting amplifier balances the photocurrent, I_{ph} , by the current through MOSFET M1 so that the voltage of the photodiode, V_{diode} , gets clamped. Due to this clamping the generated photocurrent does not need to directly modify the voltage of the read-out MOSFET M5 by a great amount. The response time of the output signal is increased so that the photoreceptor circuit can be operated under a wide range of background light conditions. And since the inverting amplifier and not the photodiode itself drives the photoreceptor output, V_{pr} , the **photoreceptor with feedback loop** can drive high loads. The response of the circuit follows:

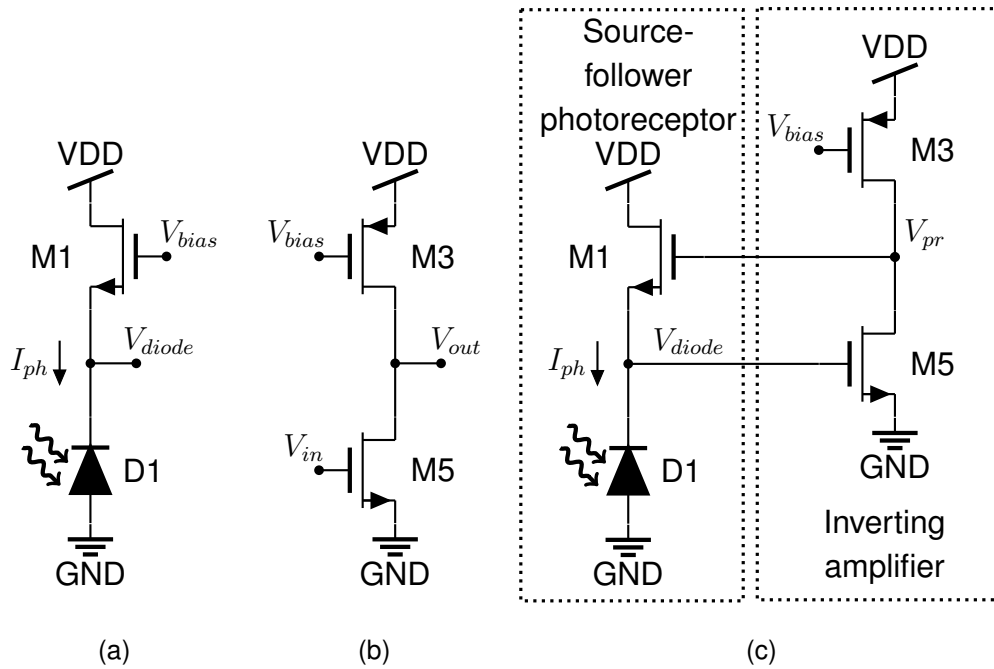


Figure 7.5: Schematics of (a) source-follower photoreceptor, and (b) inverting amplifier circuit. (c) The logarithmic photoreceptor with feedback loop combines the circuits in (a) and (b) to provide a logarithmic output voltage signal and speed up the receptor's response time.

$$V_{pr} = \frac{1}{\kappa} \left(V_{diode} + U_T \ln \left(\frac{I_{ph}}{I_0} \right) \right) \quad (7.2)$$

where I_0 is the MOSFETs' off-current, κ is a parameter describing the efficiency of the gate in determining the channel potential, and U_T is the thermal voltage.

Its sensitivity to temporal contrast can be described by:

$$\frac{dV_{pr}}{dt} = U_T \frac{A_{ia}}{\kappa A_{ia} - 1} \frac{1}{I_{ph}} \frac{dI_{ph}}{dt} \approx \frac{U_T}{\kappa} \frac{1}{I_{ph}} \frac{dI_{ph}}{dt} \quad (7.3)$$

where A_{ia} - the amplification of the inverting amplifier - is typically in the range of a couple of hundreds.

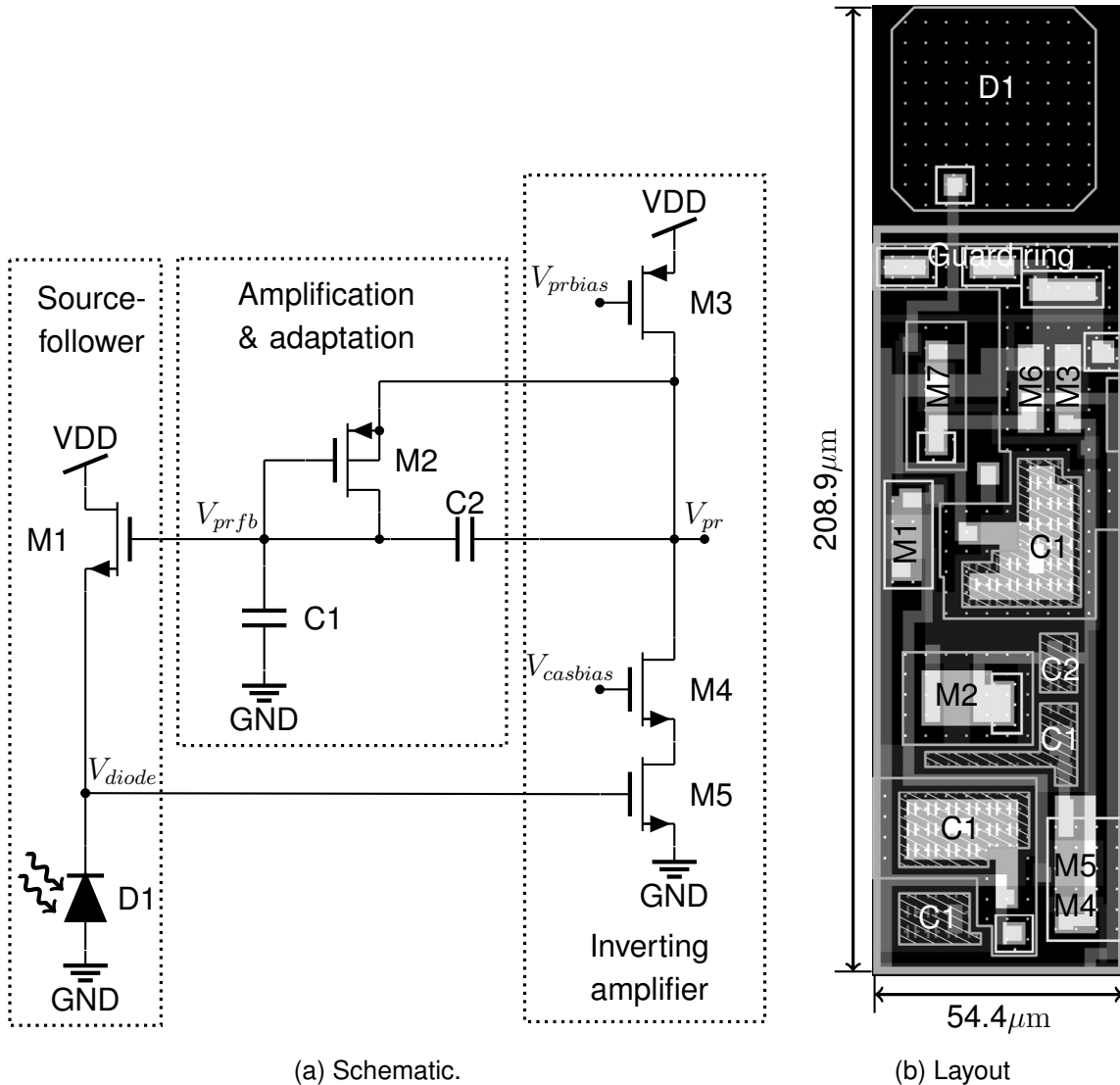


Figure 7.6: (a) Schematic of adaptive logarithmic wide-dynamic-range photoreceptor circuit by Delbrück and Mead (1994, 1995) and (b) layout. I modified the original layout by Kramer and Liu by changing the size and form of the photodiode and by adding an n-well guard ring to shield the analog MOSFETS from generated minority carriers. The layout also contains M6 and M7 from the source-follower circuit in figure 7.7. The photodiode area is $44 \times 44 \mu\text{m}^2$. C1 is implemented by four distributed capacitors with an overall capacitance of $C1=802\text{fF}$. $C2=61\text{fF}$. MOSFET sizes in μm (width/length): M1=6.4/9.6; M2=12.0/5.6; M3=5.6/9.6; M4=8.8/3.2; M5=8.8/9.6

DOFS photoreceptor circuit

By adding a capacitive divider (C1, C2), a resistive element for adaptation (M2) as well as a cascode transistor (M4) to the closed-loop photoreceptor Delbrück and Mead (1994, 1995) created the **adaptive logarithmic wide-dynamic-range photoreceptor**. This circuit also used on the DOFS is shown in figure 7.6a.

The divider composed of the capacitors C1 and C2 separates the output node, V_{pr} , from the gate of the MOSFET M1 that is driven by the voltage, V_{prfb} . By reducing V_{pr} by the factor

$$A_{pr} = \frac{C1 + C2}{C2} \quad (7.4)$$

the capacitive divider forces the inverting amplifier to respond with an amplified output voltage V_{pr} to transients in the photocurrent to allow the current through M1 to compensate the current generated in the photodiode D1 as in the photoreceptor design without capacitive divider. Since the capacitive divider by itself does not have any DC current path from the output node to the gate of M1, the photoreceptor circuit only amplifies changes in the photoreceptor output V_{pr} :

$$\frac{dV_{pr}}{dt} = A_{pr} U_T \frac{A_{ia}}{\kappa A_{ia} - 1} \frac{1}{I_{ph}} \frac{dI_{ph}}{dt} \approx A_{pr} \frac{U_T}{\kappa} \frac{1}{I_{ph}} \frac{dI_{ph}}{dt} \quad (7.5)$$

The DOFS photoreceptor circuit has a gain, A_{pr} of about 14.15.

To allow the circuit operating successfully over a large range of background light intensities Delbrück and Mead (1994, 1995) added the MOSFET M2 that acts as an resistive element allowing V_{pr} and V_{prfb} to slowly equalize over time. So the DC of V_{pr} follows the same law as the photoreceptor with feedback loop (figure 7.5c). Delbrück and Mead reported that with the adaptation through the resistive element, the photoreceptor circuit gained a total dynamic range of more than 6 decades and a dynamic range of 1-2 decades at a single adaptation level (Delbrück and Mead, 1994).

In comparison to the photoreceptor circuit shown in figure 7.5c Delbrück and Mead furthermore added a cascode MOSFET (M4) to the inverting amplifier. This cascode MOSFET nullifies the parasitic Miller capacitance between the drain and gate of M5. Without a cascode this Miller capacitance (that has its origin in the fact that the gate of every MOSFET as well forms a capacitor with the MOSFET's channel, drain and source) would allow the amplified changes in V_{pr} to couple back into the photodiode

node V_{diode} . The cascode prevents this coupling and furthermore leads to an increase in the drain resistance of M5 which increases the gain of the inverting amplifier by a factor of about 2 (Delbrück and Mead, 1995). Both effects speed up the photoreceptor's response which is useful for its application at lower light intensities.

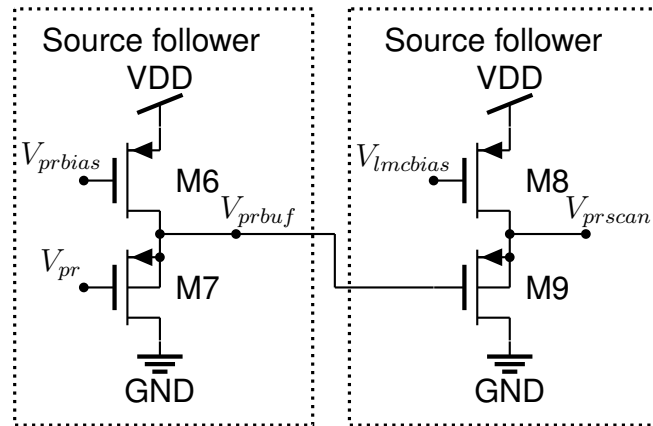


Figure 7.7: Two source follower circuits act as buffer and level shifter between the photoreceptor and LMC circuit and between photoreceptor and scanner circuit. MOSFET sizes in μm (width/length): M6=M8=5.6/9.6; M7=M9=4.8/10.4

In the DOFS, the output of the photoreceptor circuit goes through a source follower circuit before driving the LMC circuit as shown in figure 7.7. The output of the second source follower goes to the scanner. The **source followers buffer the photoreceptor output**, V_{pr} from signal changes from the scanner and LMC circuit and shifts up the V_{pr} DC voltage level. The output signal V_{prscan} is read by the scanner circuit while V_{prbuf} is fed into the LMC circuit. Since V_{prbias} and $V_{lmcbias}$ control the amount of current in the inverting amplifiers and source followers, they can be used to control the bandwidth of the circuits and to filter noise like flicker noise created by artificial light sources.

There are two main design choices that I would change if I had to redesign the layout again. These design issues were addressed in the design of the adaptive vision sensor (section 9.2.1):

1. The node V_{diode} that is unbuffered and relatively susceptible to noise is too distributed in the layout (figure 7.6b). It is implemented in the form of long wires since M1 and M5 are placed too far away from the photodiode. This is risky since other signals - especially the amplified signal V_{pr} can couple into V_{diode} .
2. The square form of the photodiode was not optimally for a linear pixel array. In a linear array that should mostly be sensitive to contrast edges traveling in the

orientation of the array, a rectangular structure as I used for my adaptive vision sensor (chapter 9) is better. A rectangular structure spatially low-pass filters contrast edges traveling orthogonal to the orientation of the pixel array more than visual stimuli traveling parallel to the direction of the pixel array.

Conclusions

Its amplification of temporal contrast support the detection of low contrast stimuli. Although it is not suitable for producing classical images, it is well suited for an optical flow sensor that should only respond to moving stimuli.

7.2.2 LMC circuit

The output of the buffered photoreceptor output V_{prbuf} is fed into an LMC circuit which provides an output signal V_{lmc} that is useful for edge detection. To detect contrast edges of small amplitude, a large photoreceptor gain, A_{pr} , (>15) is of advantage. However, A_{pr} cannot be increased without decreasing the response speed of the photoreceptor since larger gains will lead to a slower adaptation of the signal, V_{prfb} , that sets the feedback current through M1 to compensate for the photocurrent. So with high gains, A_{pr} , the clamping of the photodiode output, V_{diode} , becomes less efficient and the photoreceptor circuit loses its ability to operate under low background light intensities. Furthermore, a gain that is too high will lead to unstable behavior of the photoreceptor circuit. The LMC circuit provides additional amplification of V_{pr} .

Circuits

The LMC circuit (figure 7.8a) in the DOFS is a modified version of the original design by Liu (2000) (figure 7.9). The original circuit was part of an intensity-based correlation optical flow sensor (see section 6.3.2 for an explanation of this sensor type) and mimics the behavior of the Laminar Monopolar Cell (LMC) in the fly visual system.

The LMC circuit removes the mismatch-affected, intensity-dependent DC component of the buffered photoreceptor output signal V_{prbuf} through the capacitor C3 and provides a LMC output, V_{lmc} , which has a DC component that is only controlled through the bias voltage $V_{lmcbias}$ and that in contrast to the photoreceptor output is independent of the background light intensity. Since device mismatch also affects the behavior of the LMC circuit, the DC offset of the V_{lmc} outputs among different circuits can vary but the DC

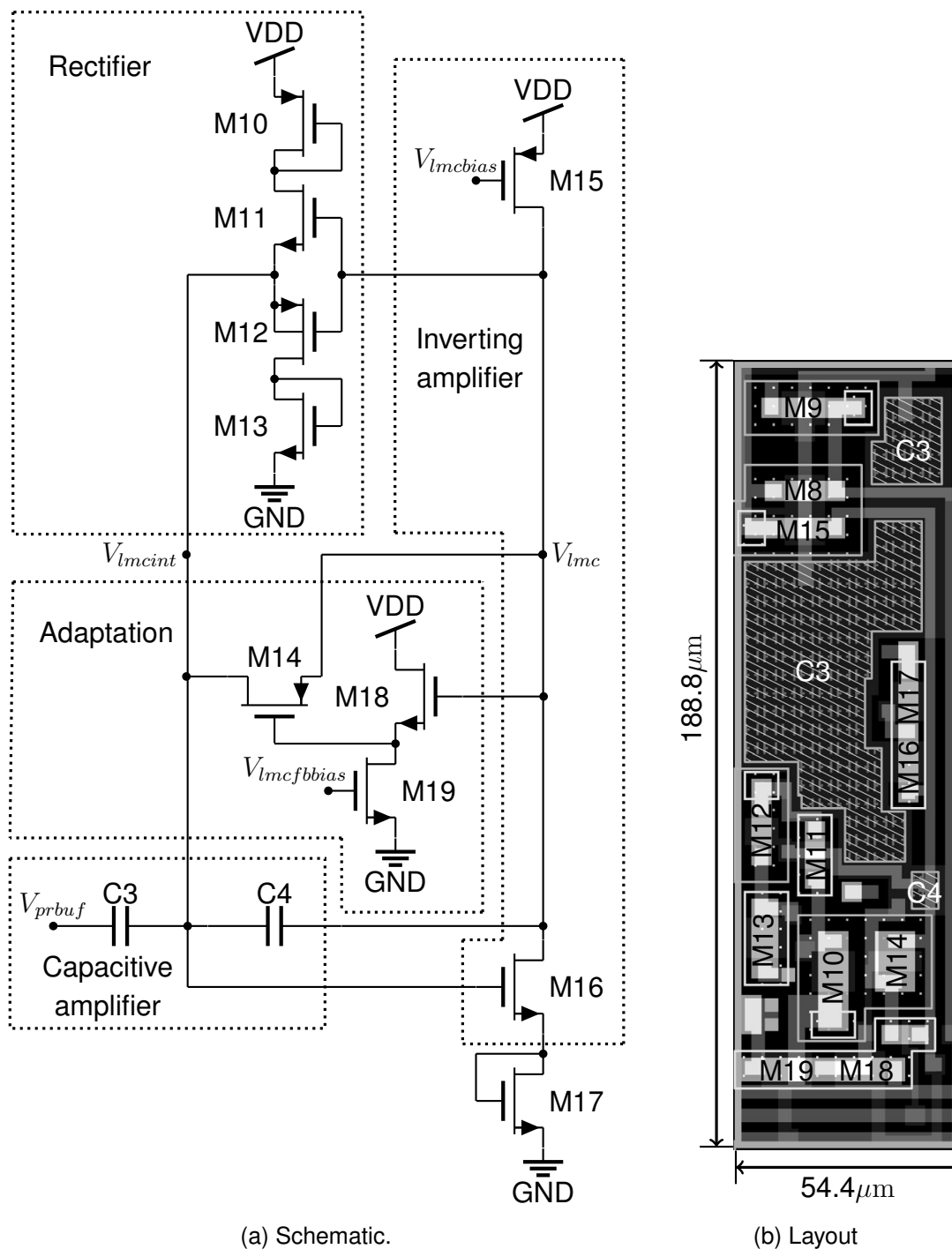


Figure 7.8: (a) Schematic and (b) layout of LMC circuit. The layout is partially taken from the unpublished design by Kramer and Liu. The layout also contains M8 and M9 from the source-follower circuit in figure 7.7. C3 is implemented by two distributed capacitors with an overall capacitance of $C3=1.49\text{pF}$. $C4=35\text{fF}$. MOSFET sizes in μm (width/length): M10=7.2/9.6; M11=M12=4.8/6.4; M13=6.4/9.6; M14=9.6/4.8; M15=M16=M17=4.8/9.6; M18=M19=5.6/12.0

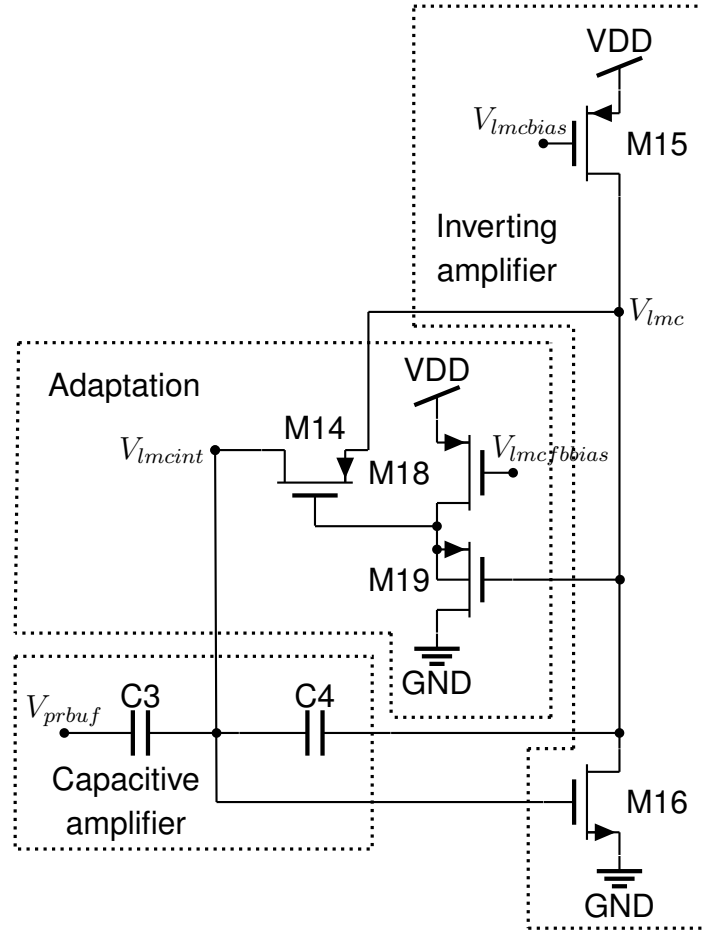


Figure 7.9: LMC circuit schematic of the original design by Liu (2000).

offset mismatch of the previous circuits (PR and buffer circuit) do not contribute to the LMC DC offset.

Both the LMC circuits by Liu and the one used in the DOFS design are based on an inverting differentiator that consists of an inverting amplifier (M15, M16), a capacitive amplifier (C3, C4) and an adaptive element M14. C3 and C4 amplify transients in the buffered photoreceptor output by the factor A_{lmc} :

$$A_{lmc} = \frac{C3 + C4}{C4} \quad (7.6)$$

so that

$$\frac{dV_{lmc}}{dt} = A_{lmc} \frac{dV_{pr}}{dt} \quad (7.7)$$

where the gain A_{lmc} in the DOFS design is about 43.57.

The total amplification of the temporal contrast due to PR and LMC circuit thus is

$$\frac{dV_{lmc}}{dt} \approx A_{lmc} A_{pr} \frac{U_T}{\kappa} \frac{1}{I_{ph}} \frac{dI_{ph}}{dt} \quad (7.8)$$

with a total gain of $A_{lmc} A_{pr}$ of about 616.1.

In contrast to the adaptive photoreceptor design by Delbrück and Mead (1994, 1995) (figure 7.6a) the LMC circuit by Liu (2000) uses an adaptive element where a single MOSFET (M14) is controlled by a source follower (M18, M19). The resistive element allows the output V_{lmc} to return to its DC level with a time constant which is set by the bias voltage $V_{lmcfbias}$. In contrast to the single-MOSFET resistive element by Delbrück and Mead (1994, 1995), the adaptive element by Liu (2000) requires three analog MOSFETs. However, at the cost of 2 extra MOSFETs, the adaptation time constant can be tuned after fabrication.

For DOFS I combined two adaptive elements - (1) the resistive element proposed by Liu (2000) (M14, M18, M19) and (2) the rectifier (M10-M13) from the unpublished sensor design by Kramer and Liu. In the unpublished design, the currents in M10 and M13 were mirrored and fed into a completely different edge detection circuit that produced contrast-dependent outputs. Instead, DOFS uses the voltage output signal V_{lmc} as the input to the edge detection circuit.

Both adaptive elements are placed in the feedback path of the inverting amplifier so that the swing of the output voltage signal V_{lmc} is limited. The current-voltage characteristics of the adaptive elements are shown in figure 7.10. Both adaptive elements act like non-linear resistors.

As can be seen in figure 7.10b the rectifier circuit provides a more symmetric response than the adaptive element by Liu (2000). This is of advantage if the LMC circuit should generate a similar response for ON and OFF temporal edges in visual stimuli. The precise current-voltage characteristic of the rectifier can be adjusted before fabrication by selecting the width-to-length ratios of the transistors M11 and M12. Since currents generated through ON edges and currents generated due to OFF edges are fed through different pathways (M11, M12) the response of the LMC circuit to ON and OFF edges can be separately adjusted by choosing the width and length of the MOSFETs M11 and M12. Furthermore, the rectifier does not make any current flow directly between the nodes V_{lmc} and V_{lmcint} supporting an increased response time to changes in the output voltage.

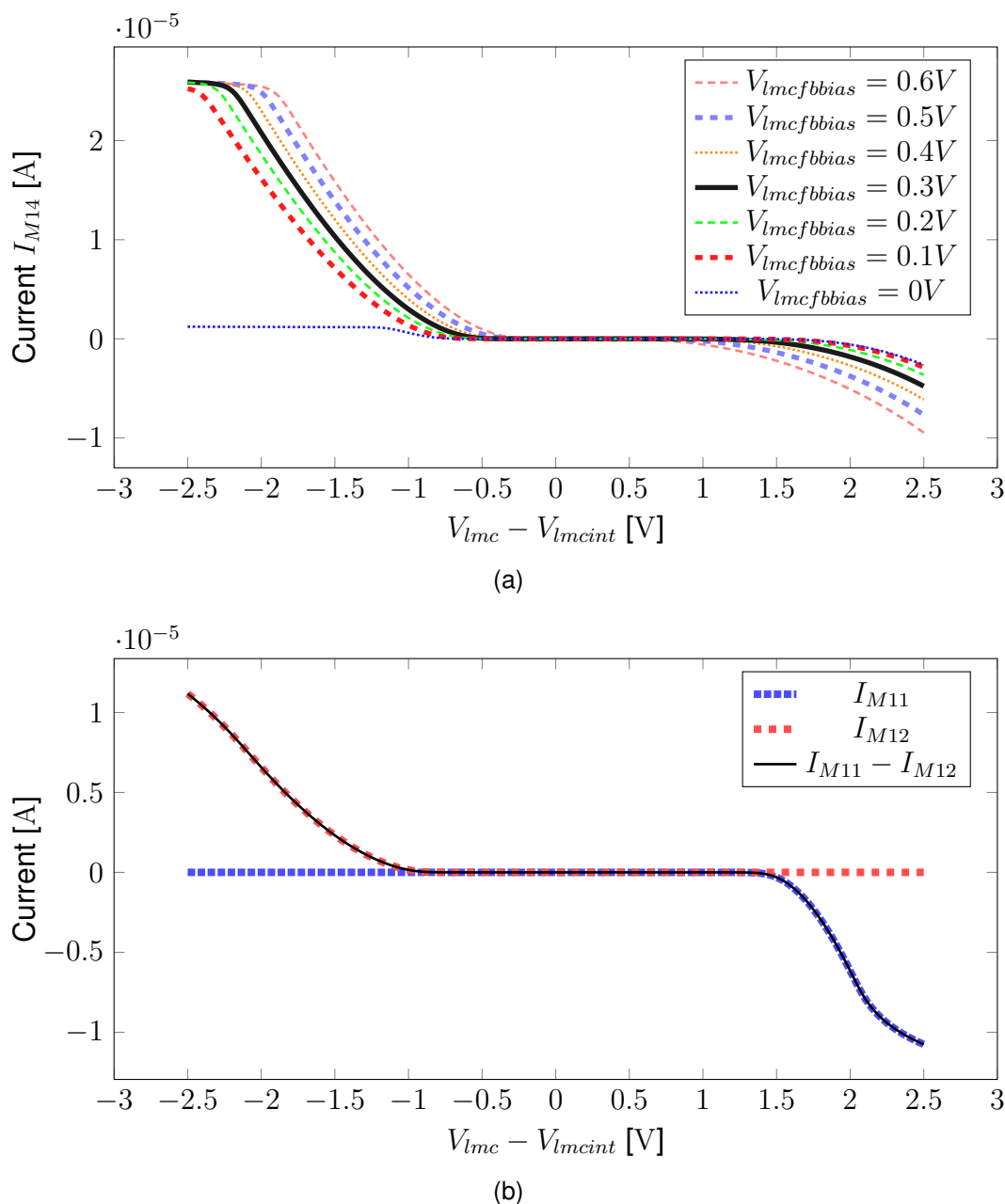


Figure 7.10: Current-voltage characteristics of adaptive resistive elements: Drain-to-source currents of transistors of (a) adaptive element by Liu (2000) for different bias voltages, and (b) rectifier. Simulation results obtained with Tanner Tools.

The adaptive element by Liu (2000) shows a more asymmetric current-voltage characteristic (Figure 7.10a). Since only one MOSFET is used to allow currents flowing in both directions directly between the nodes V_{lmc} and V_{lmcint} the asymmetric characteristic cannot be further adjusted. However the adaptive element by Liu (2000) allows to

still adjust the adaptation time constant of the LMC circuit after fabrication through the bias voltage $V_{lmcfbias}$.

For DOFS I decided to combine both adaptive elements to ideally have a symmetric response to temporal ON and OFF edges but in case it is required to keep the flexibility to adjust the circuits time constant after fabrication.

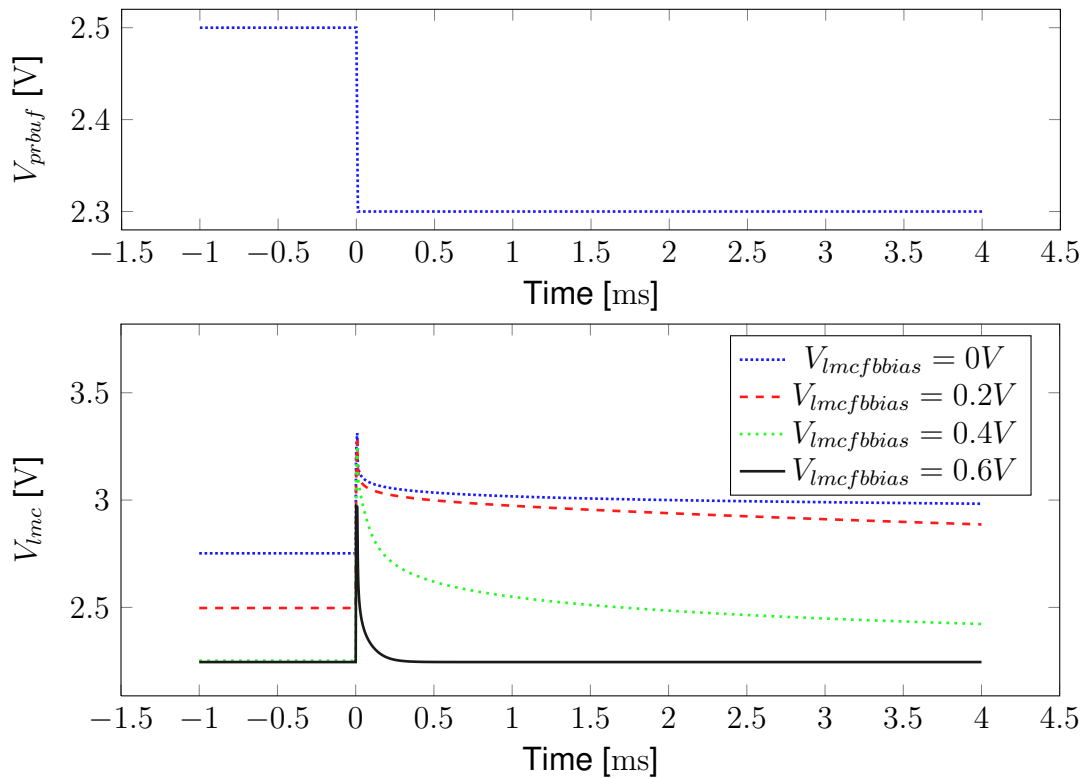


Figure 7.11: Step response of DOFS LMC circuit for different $V_{lmcfbias}$ values. Simulation results obtained with Tanner Tools.

Figure 7.11 presents simulation results on step responses of DOFS LMC circuit. The results indicate how the adaptation time of the LMC circuit changes for different $V_{lmcfbias}$ values.

Another modification to the original design by Liu (2000) is the addition of the diode connected MOSFET M17 to the inverting amplifier. M17 lifts the DC of the LMC output voltage by one diode drop (about 0.6 to 0.7V). M17 thus makes V_{lmc} better suited for comparison against the thresholds in the edge detection circuit being described in the following section.

Layout

The layout of the LMC circuit is shown in figure 7.8b. I modified the original layout by Kramer and Liu for the new circuits. In future, I would not wire the signals V_{prscan} and V_{lmc} on top of the poly1 area of C3 to reduce coupling effects and signal noise.

Conclusions

The LMC circuit for the DOFS was designed to generate output signals that are further processed by my novel edge-detection circuit. Due to the LMC circuit, temporal contrast edges in the photoreceptor output can be further amplified. The DC offset mismatch in the outputs of the photoreceptor and buffer circuits and the dependence of the offset on the background light intensity can be eliminated.

7.2.3 Double threshold edge detection circuit

The edge or token detection circuit plays a major role for the reliability of any token-based optical flow sensor. The task of the token detection circuit is the generation of an event signal that indicates the detection of a predefined token - typically a contrast edge - in the visual stimuli. To guarantee a reliable operation of the circuits implementing the optical flow algorithm from the event signal, it is critical that the properties of the event signal indicating the presence of a contrast edge are independent from the properties of the contrast edge. For example, the amplitude and slew rate of the event signal should not vary with the contrast magnitude and spatial frequency of the visual stimuli. Otherwise, the sensor's optical flow outputs will become dependent on other properties of the visual stimuli than just the stimulus velocity. Optic flow measurements that are dependent on spatial frequency and stimulus contrast can be difficult for use in the control of a closed-loop robotic platform.

From the study of the existing aVLSI token-based optical flow sensor implementations (chapter 6) I found that the contrast dependence of their optical flow outputs was mainly caused by their edge detection circuits that output event signals with properties that depend on the contrast of the visual stimuli. This is also a problem with the designs by Kramer et al. (1995, 1997). As a result, I designed a novel double threshold edge detection circuit which removes the effect of stimulus contrast.

Another aspect that needs to be considered for the design of an edge detection circuit is the generation of false events in the presence of noisy visual stimuli and circuit noise.

In contrast to intensity-based optical flow algorithms, token-based algorithms only operate when the presence of a token is clearly identified in the visual stimulus. This is an advantage since the optical flow algorithm is only executed when reliable signals are present but this also means that token-based algorithms require reliable token detection. Without reliable token detection their optical flow outputs become noisy and unreliable.

The edge detection circuit presented here uses a two-threshold scheme to reduce the effect of noise. It compares the LMC output signal against two thresholds. Both thresholds are controlled by bias voltages that can be adjusted online so that the threshold difference is large when the sensor is operated in noisy light conditions or small so that the sensor can detect low contrast stimuli under better light conditions.

Circuits

The circuit schematics and layout of the double threshold edge detection circuit are shown in figure 7.12. The circuit contains two 5-transistor differential operational transconductance amplifiers (OTA) (Comparator1: M20-M24 and comparator 2: M25-M29) that compare the LMC output signal, V_{lmc} against the two thresholds V_{thr1} and V_{thr2} . By limiting the current in the comparators with the bias $V_{otabias}$ the circuits are guaranteed to operate in the subthreshold regime. Under this condition the comparators' transconductance g_m is

$$g_m = \frac{dI_{out}}{d(V_1 - V_2)} \approx \frac{I_{bias}\kappa}{2U_T} \quad (7.9)$$

Their output conductance g_d is given by

$$g_d = -\frac{dI_{out}}{dV_{out}} \approx \frac{I_{bias}}{V_E} \quad (7.10)$$

From g_m and g_d one can compute the gain of the differential operational transconductance amplifier:

$$A_{ota} = \frac{dV_{out}}{d(V_1 - V_2)} = \frac{g_m}{g_d} \approx \frac{\kappa V_E}{2U_T} \quad (7.11)$$

which is typically in the range of a couple of hundreds. Due to this large gain, comparator 1 and 2 have nearly digital outputs.

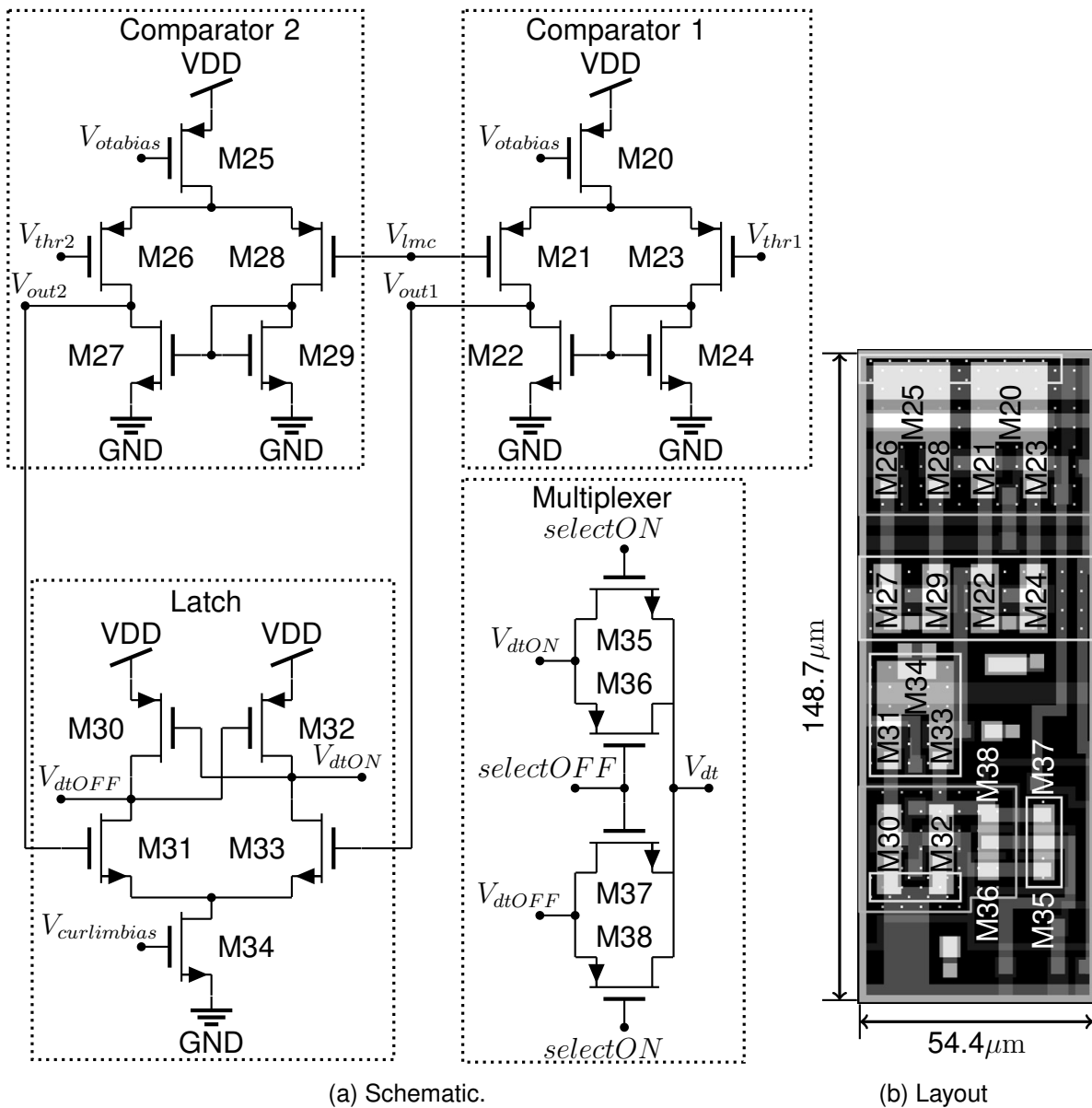


Figure 7.12: (a) Schematic and (b) layout of edge detection circuit. MOSFET sizes in μm (width/length): $M20=M25=M34=17.6/4.8$; $M21=M23=M26=M28=M30-M33=5.6/6.4$; $M22=M24=M27=M29=6.4/6.4$; $M35-M38=4.8/1.6$

Figure 7.13 depicts a timing diagram showing the signals of the edge detection circuit that have been found with the circuit simulation software from Tanner Tools. The outputs of the comparators V_{out1} and V_{out2} switch whenever V_{lmc} crosses the respective thresholds V_{thr1} and V_{thr2} .

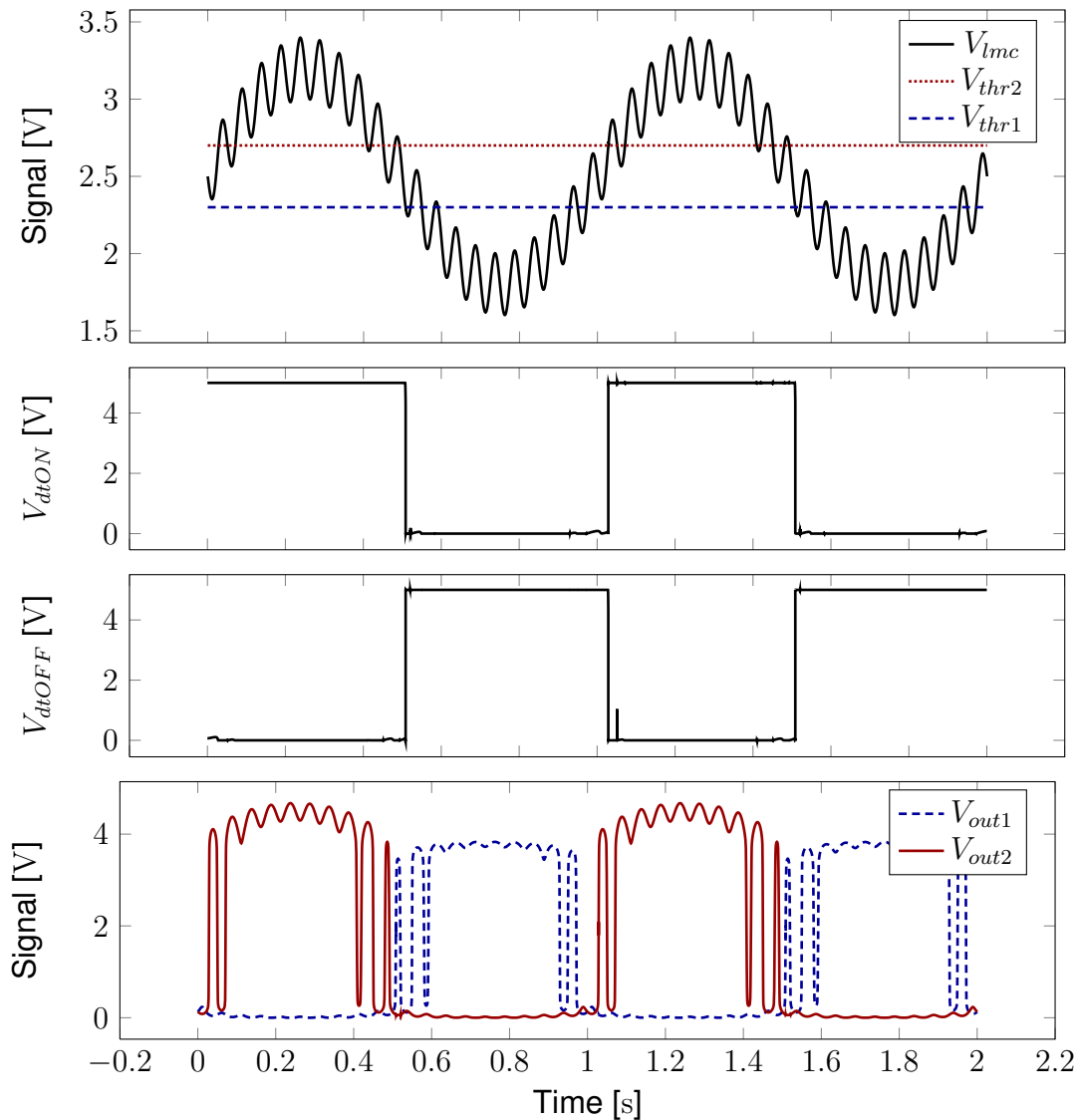


Figure 7.13: Timing diagram for edge detection circuit (Circuit simulations with Tanner-Spice). A temporal ON or OFF edge is only generated if the LMC output signal, V_{lmc} , crosses both thresholds V_{thr1} and V_{thr2} . If V_{lmc} crosses only one threshold the latch circuit ensures that the digital edge detection signals V_{dtON} and V_{dtOFF} remain constant. This way noise that causes the V_{lmc} -signal to cross a single threshold can only cause a jitter in the edge detection event but does not trigger a false edge detection event. In this example plot, V_{lmc} is a 1Hz sinusoidal signal with 1.4V peak-to-peak amplitude corrupted by a 20Hz sinusoidal noise signal with 0.4V peak-to-peak amplitude. The outputs of the comparators V_{out1} and V_{out2} switch whenever V_{lmc} crosses the respective threshold V_{thr1} and V_{thr2} . The comparator's gains are not sufficiently high to generate a fully digital output signal V_{out1} and V_{out2} .

V_{out1} and V_{out2} are used to set and reset the latch or memory circuit (M30-M34) shown in figure 7.12a. The latch circuit is composed of two cross-coupled MOSFETs (M30,M32) and a differential pair (M31,M33,M34) which maximum net current is controlled by the bias voltage $V_{curlimbias}$. The latch circuit guarantees that if both V_{out1} and V_{out2} are $\approx GND$ the states of the signals V_{dtON} and V_{dtOFF} remain unchanged. $V_{out1} = VDD$ or $V_{out2} = VDD$, however set V_{dtOFF} and V_{dtON} to VDD , respectively, while resetting the their counterpart to GND . This is why the V_{dtON} -signal generates a rising edge indicating that a temporal contrast ON edge - a transition from dark to bright light - is detected only if V_{lmc} crosses both thresholds - first V_{thr1} and then V_{thr2} . A rising edge on V_{dtOFF} that signals the detection of a temporal contrast OFF edge - a transition from bright to dark light - is only generated at the crossing of V_{thr2} if V_{lmc} passed V_{thr1} before.

The implementation of the two thresholds in combination with the latch circuit ensures that small signal variations in V_{lmc} cannot trigger additional rising edges in the edge detection signals V_{dtON} and V_{dtOFF} . The distance between the thresholds allows one to select the amplitude of noise and signal contrast that should be rejected. If the distance between the threshold is small, low contrast stimuli and small variations in V_{lmc} can trigger edge detection events while a large difference between the thresholds guarantees V_{dtON} and V_{dtOFF} to change their state only for large changes of V_{lmc} .

Besides the comparators and latch circuit, the edge detection circuit includes a multiplexer composed of two transmission gate circuits (M35-M38). This multiplexer allows for selecting either ON or OFF events to trigger the subsequent pulse shaping circuit and ensures that ON and OFF contrast edges are not mixed during the optical flow measurement.

Figure 7.14 shows the V_{dt} event signals generated by the edge detection circuit when being stimulated with sinusoidal V_{lmc} input signals of different frequencies. The circuit generates sharp ON-edges for input frequencies between 0.1Hz and 100kHz. This is more than the range of frequencies expected in the targeted autonomous robots. Below 0.1Hz the rising edge of the output signal suffers from the limited gain of the input comparators. However the falling edge of the edge detection event is not effected due to the additional gain of the latch circuit. For input signal frequencies above 1MHz V_{dt} becomes rounded due to the limited gain of the circuit.

Conclusion

The edge detection circuit is an important core element of the DOFS. It helps to reduce the dependence of the flow output to variations in stimulus contrast, spatial frequency

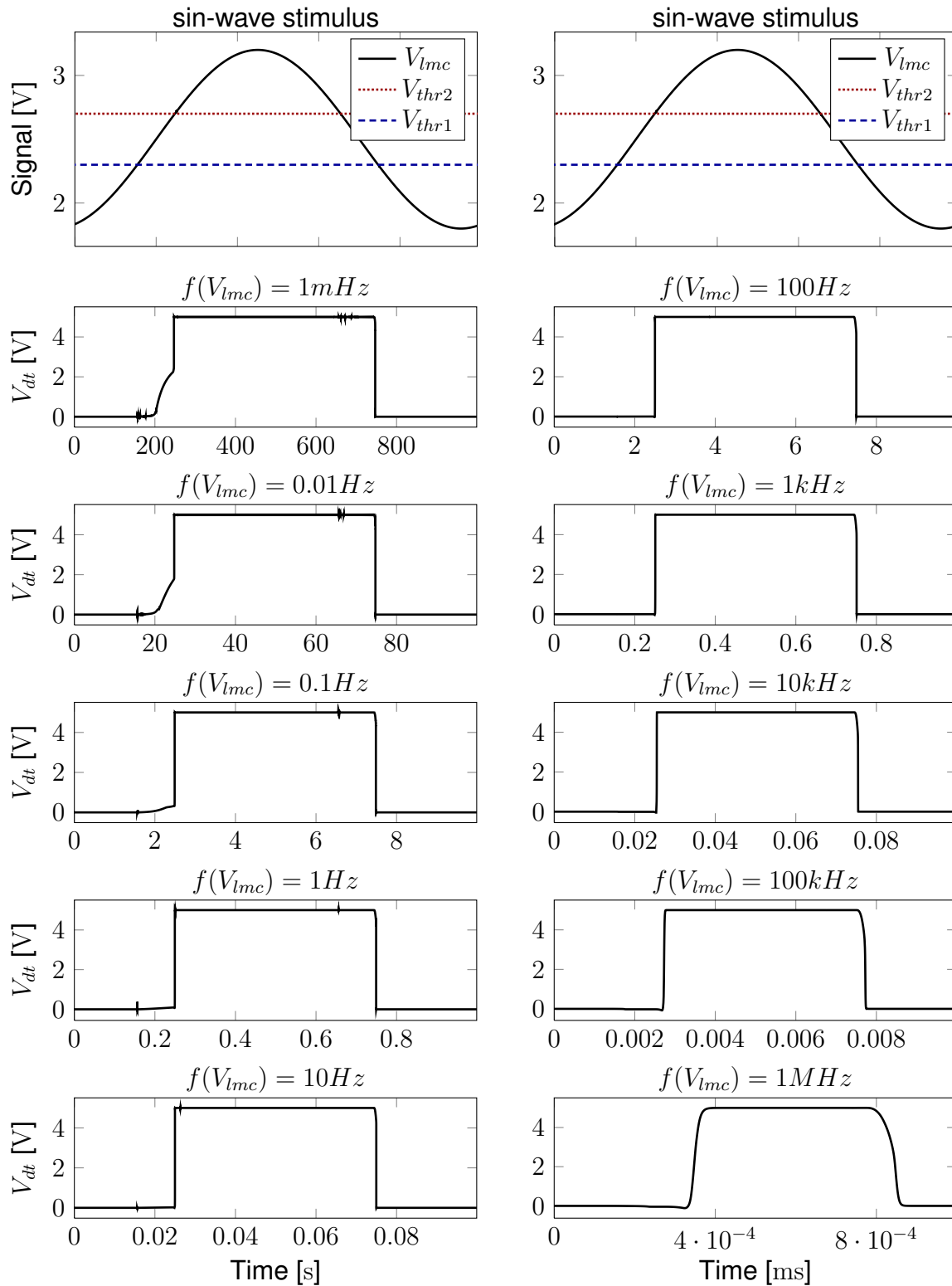


Figure 7.14: Output events, V_{dt} , from edge detection circuit when the circuit is stimulated with sinusoidal V_{lmc} -signals with varying frequencies and constant peak-to-peak amplitude of 1.4V. The thresholds V_{thr1} and V_{thr2} have been set to 2.3V and 2.7V, respectively. (Simulation results generated with Tanner-Spice.)

and noise. The application of this edge detection circuit is not limited to this facilitate-and-sample algorithm. In fact, the combination of photoreceptor, LMC and edge detection circuit can form a robust front-end for implementations of any of the token-based algorithms described in chapter 6 in analog VLSI technology - no matter if it is for an implementation of the correlation algorithm (section 6.3.3) or one of the various time-of-travel algorithms (section 6.3.4).

7.2.4 Circuits for pulse shaping and generation of facilitate signal

Together with the following sample & hold & decay circuit, the circuits for pulse shaping and generation of the facilitate signal implement the actual facilitate-and-sample optical flow algorithm.

The circuits for pulse shaping and generation of the facilitate signal are shown in figure 7.15. The circuits for pulse shaping generates a digital pulse P of controlled width in time from a rising edge in the edge detection circuit output signal V_{dt} . As depicted in the timing diagram in figure 7.15 a delay circuit (M39-M44) generates a delayed signal $V_{dt\text{delay}}$ in response to a rising or falling edge in V_{dt} . The delay of the rising and falling edges can be individually controlled via the biases $V_{pw\text{bias}}$ and $V_{hys\text{bias}}$, respectively, that limit the currents in the starved inverter circuit (M39-M42) of the delay circuit.

An OR gate compares the delayed ($V_{dt\text{delay}}$) and inverted (V_{ndt}) version of the edge detection signal and generates a pulse nP which pulse width only depends on the bias $V_{pw\text{bias}}$. Via the bias $V_{hys\text{bias}}$ a refractory period - a time in which no additional pulse can be generated - can be set. The refractory period is an additional mechanism in the DOFS design to avoid the generation of false pulses due to circuit noise since it prevents the accidental generation of bursts of pulses. I included it in the design but finally never actively used the refractory for filtering noise since the double threshold mechanism of the edge detection circuit was already so affective.

The generation of the facilitate signal is implemented by two MOSFETS (M55, M56) and a single capacitor (C5). Whenever a pulse nP is generated in response to a detected contrast edge the capacitor C5 is charged to the voltage level of $V_{srcpbias}$. C5 is discharged by a constant current source (M56) to the voltage level $V_{srcnbias}$ so that the voltage level on C5 - the facilitate signal F - linearly decreases over time from $V_{srcnbias}$ to $V_{srcpbias}$. In practice $V_{srcpbias}$ and $V_{srcnbias}$ can be set to VDD and GND, respectively. However, since the DOFS is implemented in a 5V technology it can be advantageous to connect $V_{srcpbias}$ to a 3.3V supply so that the optical flow output can be directly connected to a standard 3.3V microcontroller with integrated analog-to-digital converter.

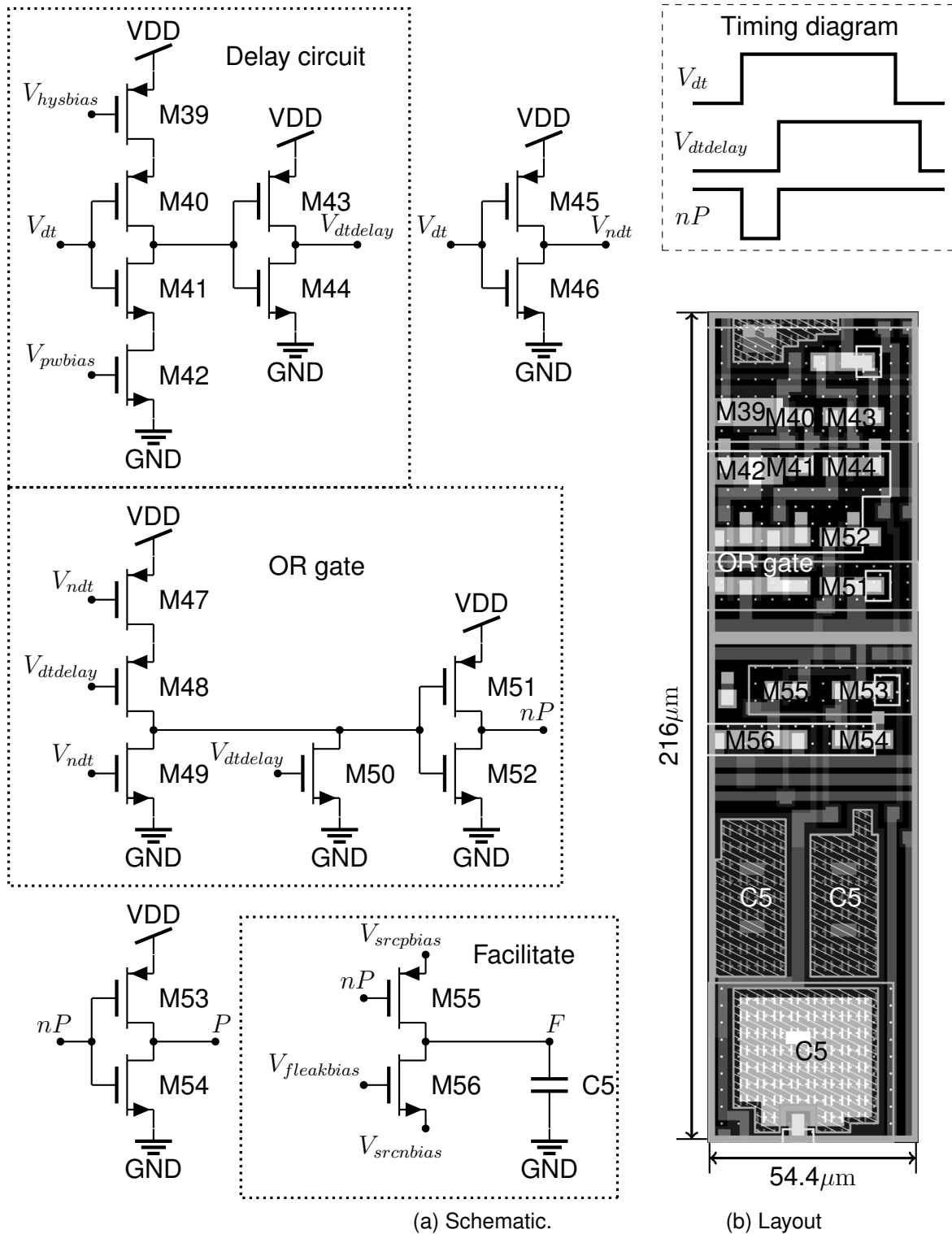


Figure 7.15: (a) Schematic and (b) layout of circuits for pulse shaping and generation of facilitate signal. C5 is implemented by three distributed capacitors with an overall capacitance of $C5=0.77\text{pF}$. MOSFET sizes in μm (width/length): $M39=M42=7.2/10.4$; $M56=4.8/9.6$; $M40=M41=M43-M55=4.8/1.6$

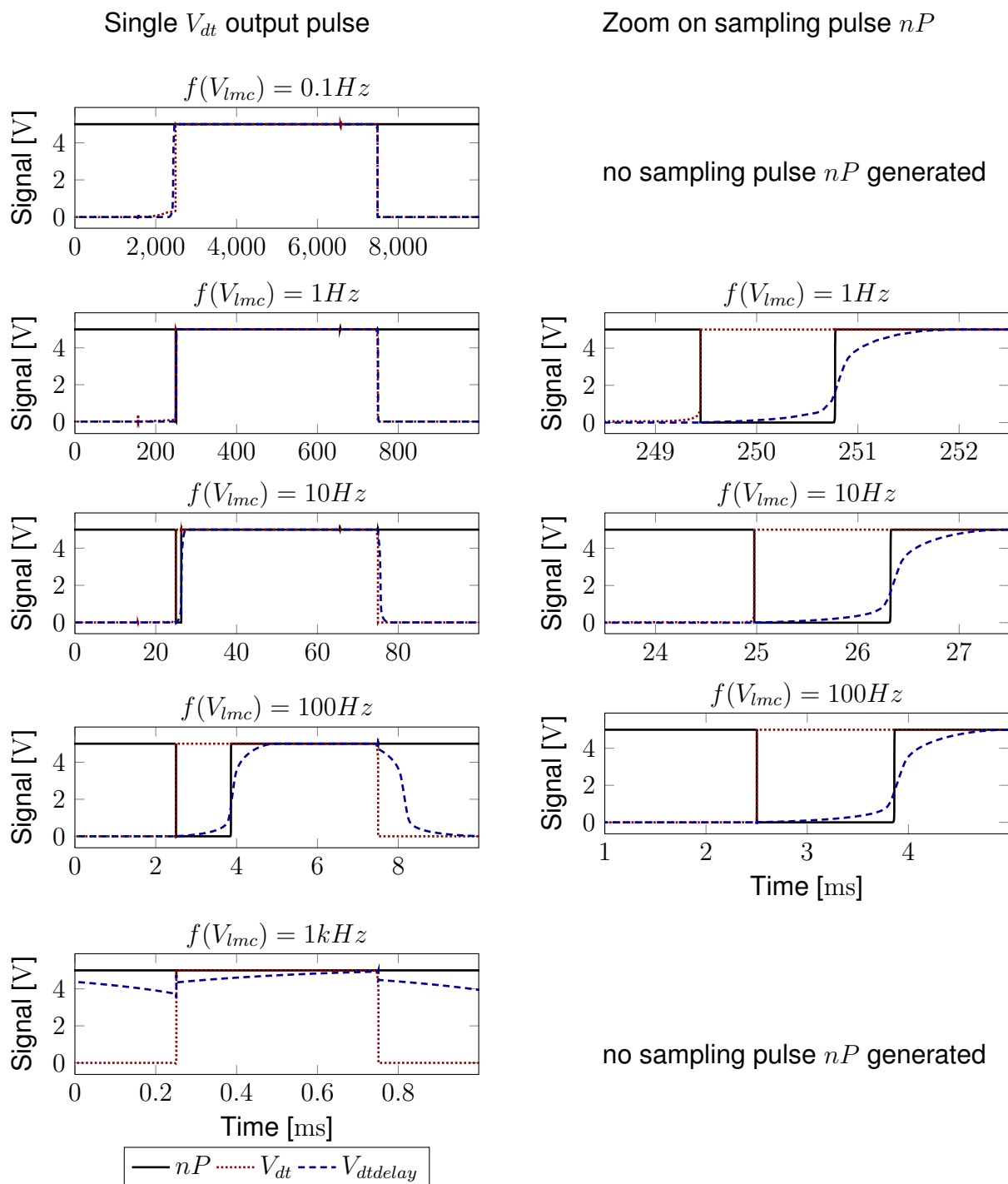


Figure 7.16: Output signal from the edge detection circuit, V_{dt} , its delayed version, $V_{dtdelay}$, and the sampling pulse, nP , generated when the input to the edge detection circuit, V_{lmc} , was stimulated with sinusoidal signals of different frequencies. The circuits generate a pulse of fixed width over at least 3 decades of input frequencies. (Simulation results with Tanner-Spice.)

Figure 7.16 shows sampling pulses nP generated in simulation while stimulating the double threshold edge detection circuit with different frequencies and feeding its output V_{dt} into the pulse shaping circuit. The circuits' bias values have been selected according to the values used with the fabricated sensor. In its present configuration the pulse shaping circuit generates pulses with constant pulse width for frequencies between 1Hz and 100Hz. Pulses for lower frequencies could be generated as well if the falling edge of the V_{dt} -signal was used instead of its rising edge. However, for the targeted flying platform it is not expected that such low frequencies should be detected. Instead stimuli with temporal frequencies below 1Hz will be filtered through the adaptation mechanism in the PR and LMC circuits. In theory the edge detection and pulse shaping circuits could also generate sampling pulses for frequencies above 100Hz. In practice although the edge detection circuit generates events under the chosen bias values the edge detection circuit does not. As shown in figure 7.16 the sampling pulse width was chosen to be $1.35ms$ to ensure that the following sample & hold & decay circuit properly samples the facilitate signal for all the fabricated pixels of the optical flow sensor despite fabrication mismatch between individual pixels. This pulse width does not allow the generation of pulse for frequencies above 740Hz since the refractory period prevents the generation of additional pulses once a single pulse is generated.

7.2.5 Sample & hold & decay circuit

The sample & hold & decay (SHD) circuit samples and stores the voltage of the facilitate signals from the right (F_R) and left (F_L) neighbor of an optical flow pixel using the pulse P .

The circuit and its layout are shown in figure 7.17. The SHD circuit is a standard design as it has been used by Kramer et al. (1995) and others before. It comprises two unity-gain follower circuits (M57-M61 and M64-68) composed of 5-transistor differential transconductance amplifiers which outputs are coupled to their negative input terminal, respectively. The unity-gain followers provide an output signal V_{out} according to

$$\frac{dV_{out}}{dV_{in}} = \frac{A_{ota}}{A_{ota} + 1} \approx 1 \quad (7.12)$$

where A_{ota} is the gain of a standard 5-transistor operated in the subthreshold domain (7.11). The unity-gain followers buffer the respective facilitate signals and allow for a fast charging of the storage capacitors C6 and C7 that hold the sampled facilitate signals, H_R and H_L . C6 and C7 are charged through the MOSFET switches M62 and M69 that

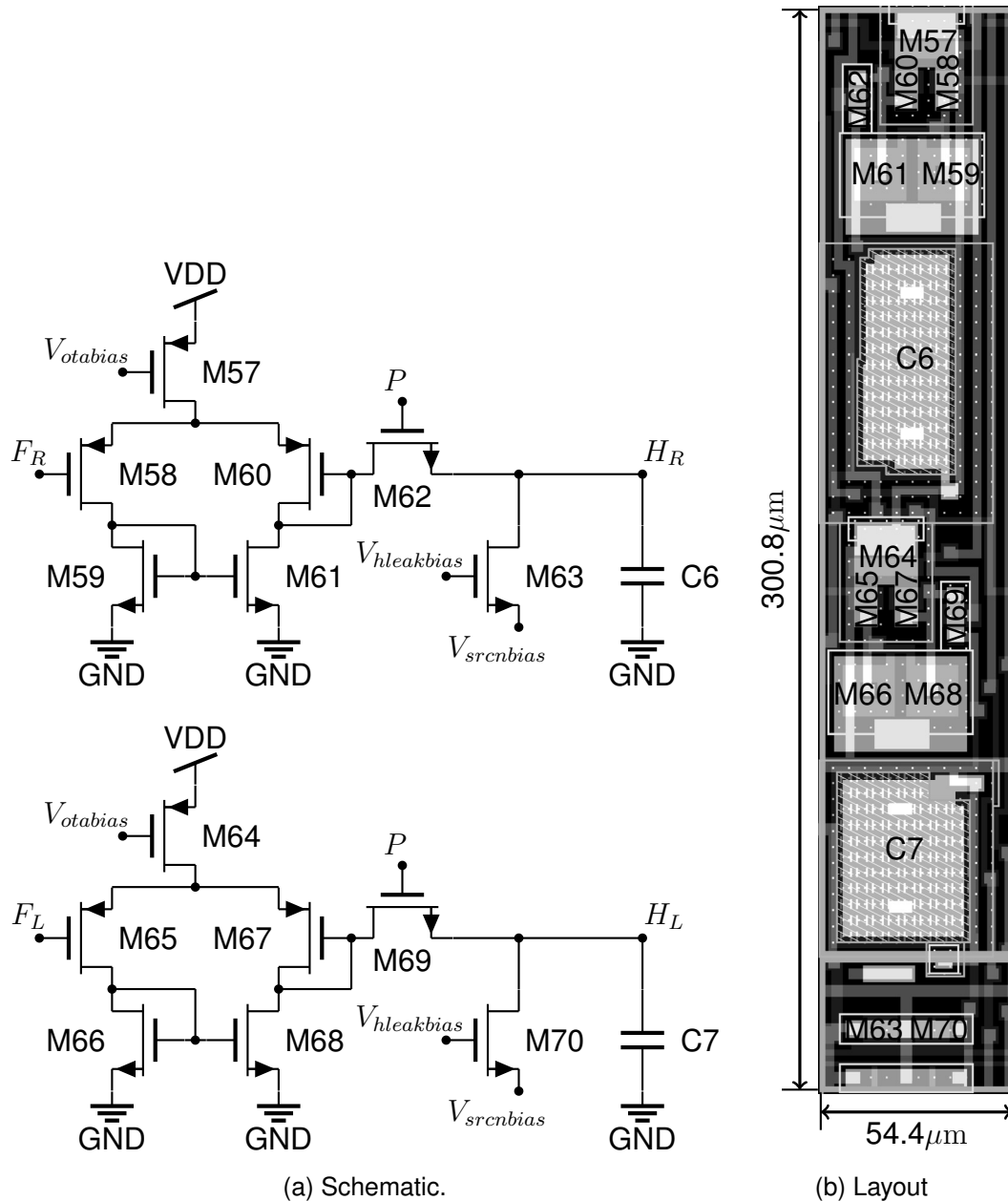


Figure 7.17: (a) Schematic and (b) layout of sample & hold & decay circuit. $C6=C7=1.0\text{pF}$. MOSFET sizes in μm (width/length): $M57=M64=17.6/4.8$; $M58=M60=M65=M67=6.4/6.4$; $M59=M61=M66=M68=19.2/4.8$ (implemented as ring transistors); $M62=M69=4.8/6.4$; $M63=M70=4.8/9.6$

are activated by the pulse P . The pulse width of P thus has to be long enough to allow charging $C6$ and $C7$ reliably to the voltage levels of F_R and F_L . On the other hand the pulse width should be as short as possible since it limits the maximum stimulus velocity

that can be measured.

Both H_R and H_L are linearly decaying with a small time constant set by the bias voltage $V_{leakbias}$ towards the voltage level set by $V_{srcnbias}$. This leakage can be used to encode the time since the last update of the hold signals H_R and H_L and allows for an efficient encoding of a reliability signal.

7.2.6 Scanner circuit

The analog scanner circuit (Mead and Delbruck, 1991) consists of a row of shift registers that allows access to the analog signals of individual optical flow pixels. As shown in figure 7.18a, the scanner on DOFS is composed of six unit-gain followers (OTA1 - OTA6) that buffer and output the individual pixel signals when activated by one register.

The unity-gain follower can be implemented by a 2-transistor source follower circuits or through an operational transconductance amplifiers (OTA). For the scanner, I decided to use the same 5-transistor unity gain followers as the buffers in the SHD circuit (section 7.2.5) but with an additional 6th MOSFET that allow switching off the buffer. The OTA circuit is shown in figure 7.19. Although OTAs require more transistors I avoided the use of 2-transistor follower circuits since the OTAs do not provide buffered output signals with shifted offsets.

Each of the six OTAs in a pixel, buffers and allows access to one of the following signals: the LMC circuit output, V_{lmc} , the photoreceptor output as well as those of its left and right neighbor V_{pr} , V_{prL} , and V_{prR} , and the sampled and hold facilitate signals H_R and H_L . If the OTAs of several pixels are activated, the scanned output signals of these OTAs will be averaged.

During normal operation of the scanner circuit, the registers ensure that only one pixel is selected at a given time. The circuits of the register are shown in figure 7.20. This standard text-book register is based on transmission gates (M1, M2, M6, M7) that shift the voltage level from the differential input ($BITIN$, $nBITIN$) to the differential output ($BITOUT$, $nBITOUT$) at each cycle of the clock signal CLK . Two latch circuits (M10-M14 and M15-M19) store the current state of the bit values during a shift operation. The registers of the individual pixels form a shift register where during each clock cycle, the bits are shifted from each register to the following register. All registers are connected by a wired OR signal that makes sure that a new active bit is only loaded into the first register if all registers are empty.

The advantage of the present scanner design is that it allows accessing the local signals

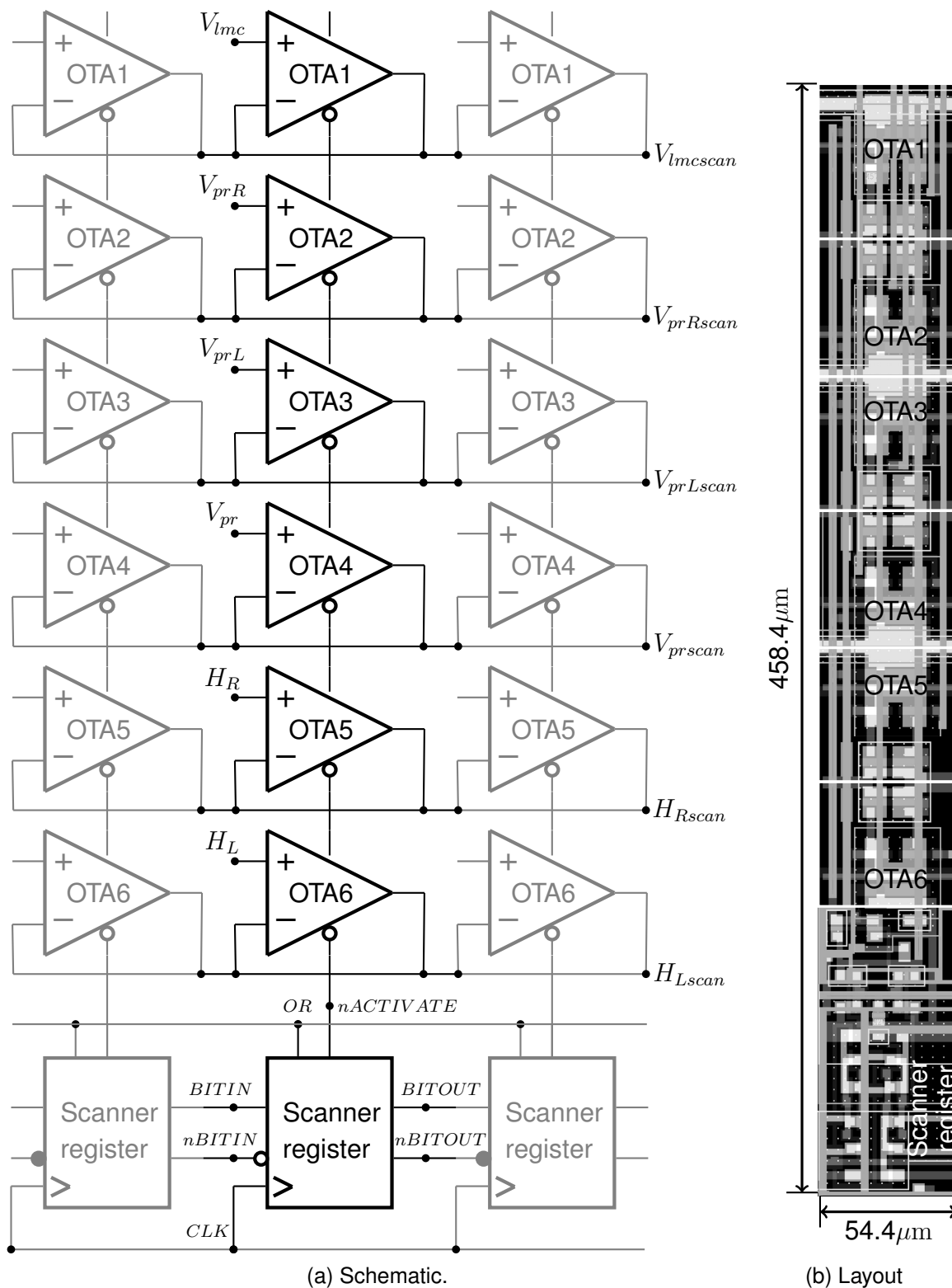


Figure 7.18: (a) Schematic and (b) layout of scanner circuit including scanner register and six OTAs for selecting and buffering local pixel output values.

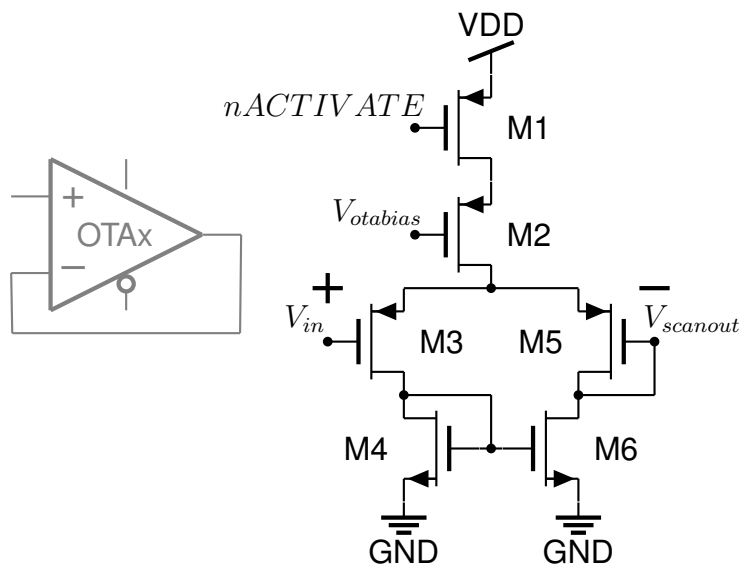


Figure 7.19: The buffer of the DOFS scanner is the well-known 5-transistor operational transconductance amplifier (M2-M6) with an additional MOSFET M1 that acts as a digital switch and allows deactivating the output of the buffer circuit.

of individually pixels easily by scanning one pixel after the other with an interface that requires only a clock signal, reading a synchronization bit to know when the shift register is empty and starts scanning again the first pixel and reading as many analog values as should be scanned. Since the scanner circuit has no reset - a choice made to save an additional input signal - one has to empty the scanner's shift register after the sensor is first powered up, by toggling the clock signal, CLK , as many times as there are pixels in the sensor.

Although the OTAs only consume power when activated, the scanner circuit is not exactly low power. This is because the register circuit only uses either an nFET or a pFET as transmission gates that without their complements do not well propagate either high or low signals. Because of the transmission gates, the register requires the inverters (M3, M5 and M8-M9, M20-M21) for restoring the voltage level and to properly drive the OR and $nACTIVATE$ signal. These inverters as well as the latches can consume high power when their input voltage level is too far from either VDD or GND. The advantage of the present register design is that it does not require the generation of an inverted clock signal.

The scanner circuit occupies a relatively large area. In fact it takes about 30% of the pixel area. This is because the scanner layout was not optimized for space and since I made the MOSFETs for the OTAs relatively large to reduce the effect of mismatch.

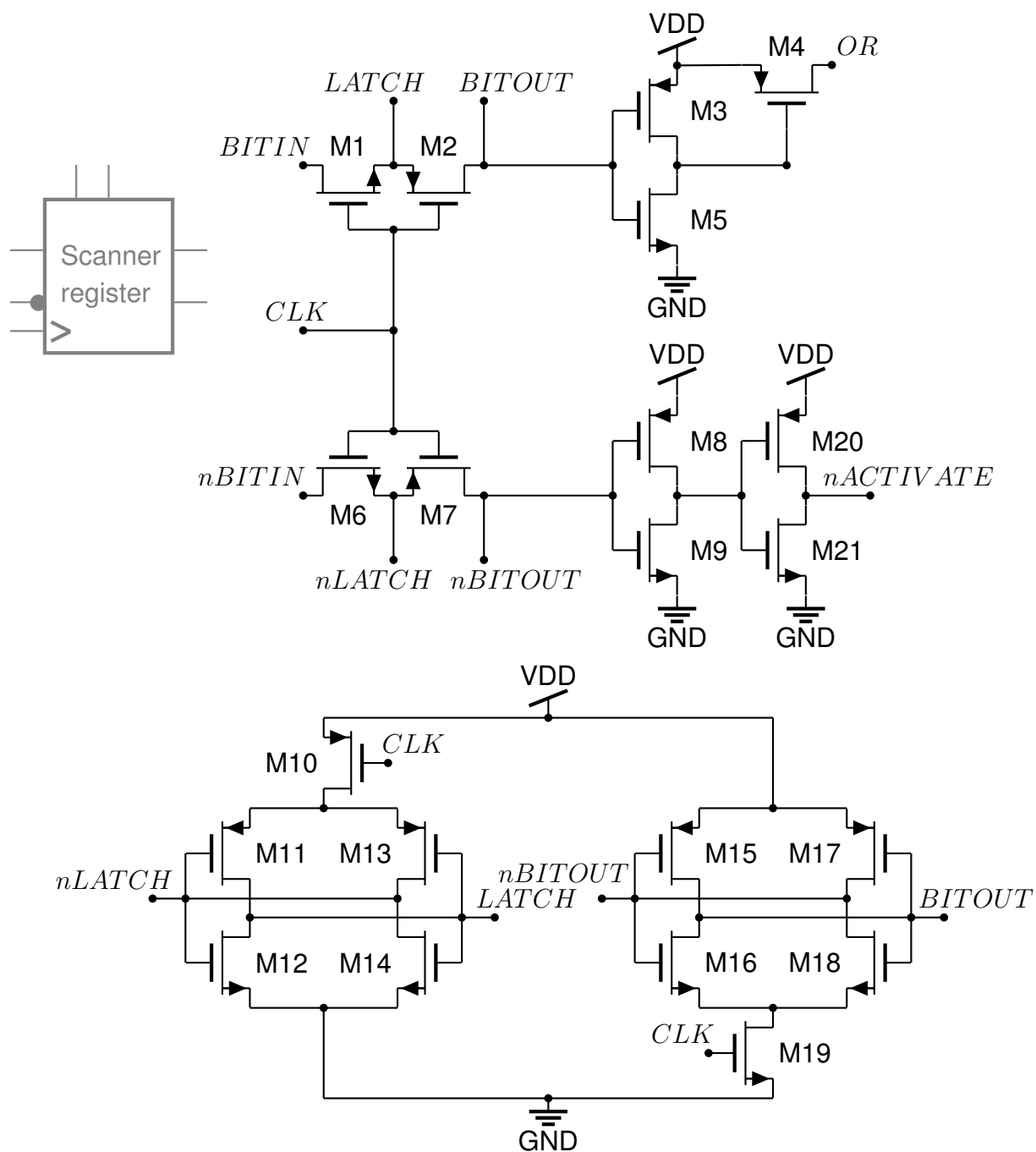


Figure 7.20: Schematic of scanner shift register.

I chose to keep the scanner circuit in the DOFS design since it is a well tested reliable circuit for analog test circuits. However, the scanner is also currently the main bottleneck for the readout speed of individual pixel values. For a small array of only 24 pixels, this is not a problem but for large pixel arrays it will become a limiting factor. So for future

designs I would prefer adding a communication interface that takes advantage of the event-based nature of the facilitate-and-sample algorithm or at least a scanner circuit where pixels can be selected through an address.

7.2.7 Direction selection circuit

The schematic and layout of the direction selection circuit are shown in figure 7.21. The circuit compares the sampled and hold signals H_R and H_L from the neighboring left and right optical flow pixel. The higher hold signals is selected as the optical flow output OF_R and OF_L and the corresponding direction flag DIR_R or DIR_L is set to VDD. The other optical flow output of the lower signal H_R or H_L is set to be equal to $V_{srcnbias}$ and direction flag is set to GND. All circuits are operated in subthreshold domain due to the bias $V_{otabias}$. The bias $V_{drainbias}$ allows to adjust the hysteresis for the comparison of the signals H_R and H_L . It avoids oscillations of the direction selection flags and optical flow outputs.

In the original design by Kramer and Liu the direction selection circuit was part of every optical flow pixel. To reduce the pixel size and to avoid additional mismatch in the optical flow outputs I decided to place this circuit in my sensor design after the scanner circuit. This way the hold signals of all local optical flow pixels are processed by the very same circuit.

However, since the direction selection circuit is vulnerable to offset mismatch in the hold signals for my experiments I typically ignore the outputs of the direction selection circuit. Instead I directly use the hold signals H_R and H_L , convert both by an analog-to-digital converter and compare their values in software on a microcontroller. The reason is that differences in the offset of the signals H_R and H_L can lead the direction selection circuit to select for the wrong direction. In software, however, offsets can be removed before comparing H_R and H_L .

7.3 Experimental results

This section presents experimental results from the fabricated dynamic optical flow sensor. The experimental results have been partially published in form of conferences articles (Moeckel and Liu, 2007, 2008) and in a book chapter (Moeckel and Liu, 2009).

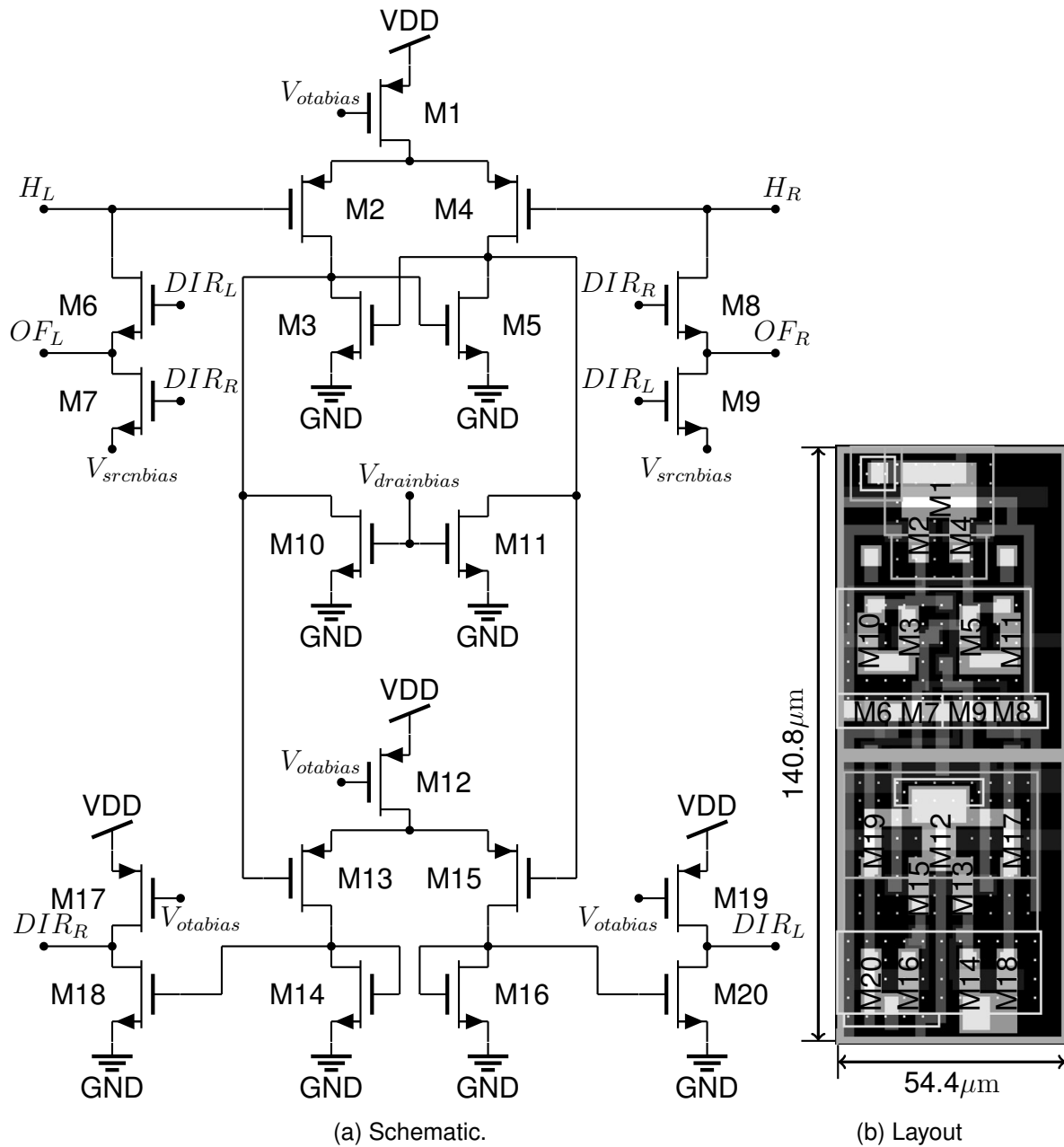
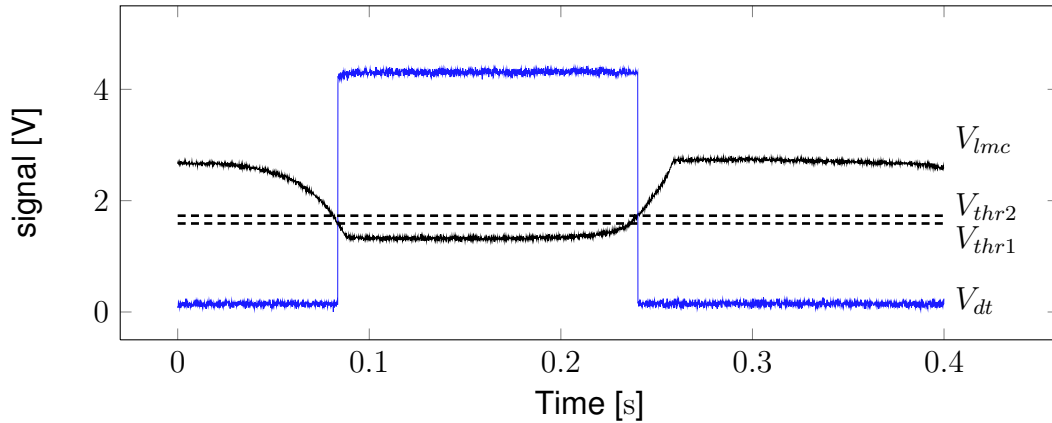


Figure 7.21: (a) Schematic and (b) layout of direction selection circuit. MOSFET sizes in μm (width/length): M1=17.6/4.8; M2-M6=M8=M13-M16=M18=M20=4.8/4.8; M7=M9=4.8/6.4; M10=M11=4.8/3.2; M12=8.0/4.8; M17=M19=4.0/4.8

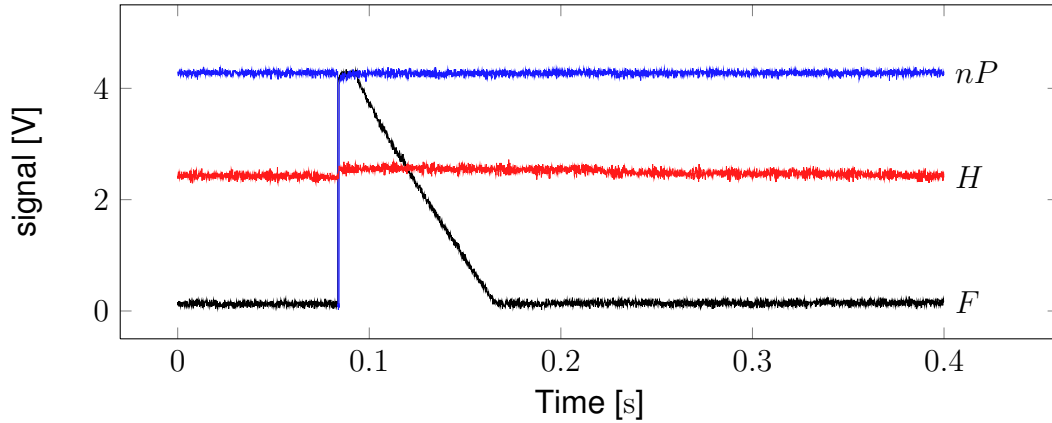
7.3.1 Measured outputs at a pixel

The different output signals from a pixel are shown in figure 7.22. The feature detection signal V_{dt} is generated once the LMC output V_{lmc} has crossed from threshold 2 to

threshold 1. The positive edge of V_{dt} is used to create the sampling pulse nP that also resets the voltage F on a capacitor. After the reset, F leaks away linearly. The pulse nP is also used to sample the F -signal from the left and right neighbors. The hold signal H shows the larger of these two sampled values.



(a)



(b)

Figure 7.22: (a) Measured outputs from the fabricated DOFS. The LMC output V_{lmc} is compared against the two thresholds. When this output crosses both thresholds, V_{dt} is high showing that a temporal OFF contrast edge is detected. This results in the generation of the sampling pulse nP and the facilitate signal. nP is used to sample the F signal from the neighboring pixels which results in the update of H . Figures adapted from Moeckel and Liu (2007).

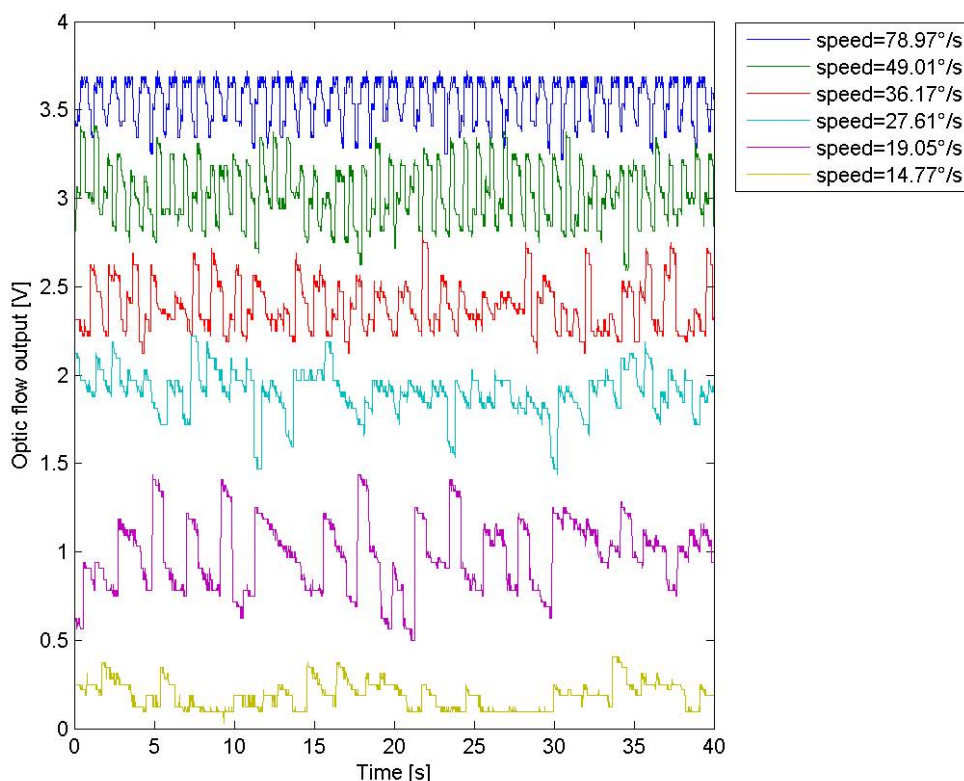


Figure 7.23: Optic flow output samples from pixel number 10 for 6 different stimulus velocities. Sharp positive updates in the signal reflect when a feature was detected. Differences in the update value reflect both stimulus and circuit noise.

7.3.2 Optic flow output from a single pixel

The 24 optical flow outputs of the DOFS were recorded when presented with a variety of moving spatial patterns on the LCD display. The optical flow sensor results were obtained while using sinusoidal patterns of 35% contrast with a spatial period of 3 cm . The sensor perceived the visual stimuli through a 6 mm focal length lens. The distance between the chip and the display was approximately 28 cm . The stimulus velocity was varied between 10°/s and 100°/s in steps of 5°/s.

Samples of raw optical flow outputs from the sensor's pixel number 10 for six different stimulus velocities are shown in figure 7.23. Each optical flow signal shows the behavior where the output is periodically updated and otherwise decays linearly over time. Since the spatial frequency of the stimulus was not changed between experiments but only its velocity, the optical flow outputs get more often updated for fast moving stimuli. For

an ideal sensor and experimental setup the optical flow outputs should be reset to the same analog voltage value at each update. In practice the sensor suffers from noise in the visual stimuli due to the screen as well as from circuit noise. This is why there is some noise in the optical flow updates as well. However the six different stimulus velocities can be clearly distinguished without the application of any filtering techniques. When low-pass filtering the optical flow outputs the quality of the signals can be further improved.

7.3.3 Optic flow output from all pixels

In the following experiments, 500 optical flow output values were recorded and averaged for each stimulus velocity and pixel. Figures 7.24 and 7.25 show the mean optical flow output profiles from the circuits that are selective for stimuli moving from the right to the left as recorded for all 24 individual optical flow pixels. Figures 7.26 and 7.27 show the optical flow output profiles from the circuits that are selective for stimuli moving from the left to the right as measured from all 24 individual optical flow pixels. Error bars indicate the minimum and maximum (figures 7.24 and 7.26) and standard (figure 7.25 and 7.27) deviation. The deviations from the mean are caused both by circuit and power supply noise in addition to noise in the moving stimulus pattern due to the passive LCD screen. When operating an optical flow sensor on a flying platform similar deviations have to be expected due to vibrations of the MAV.

Averaged optical flow output profiles across the 24 optical flow pixels as well as the mean optical flow profiles of individual pixels are shown in figures 7.28 and 7.30. The deviations between the response of individual pixels is probably caused by fabrication mismatch. The 3D plots in figures 7.29 and 7.31 show a sinusoidal mismatch pattern across the pixel array. This mismatch type that shows itself as a sinusoidal variation of transistors properties being distributed in space with slowly varying frequency is typically referred to as the striation effect (section 5.2.1).

Figures 7.32 to 7.35 depict the converted pixel output profiles $s = 1/(VDD - output)$. As theoretically predicted, these plots show a typically approximately linear relationship between s and stimulus speed. Deviations from the linear relationship are mostly caused by saturation effects and fabrication mismatch.

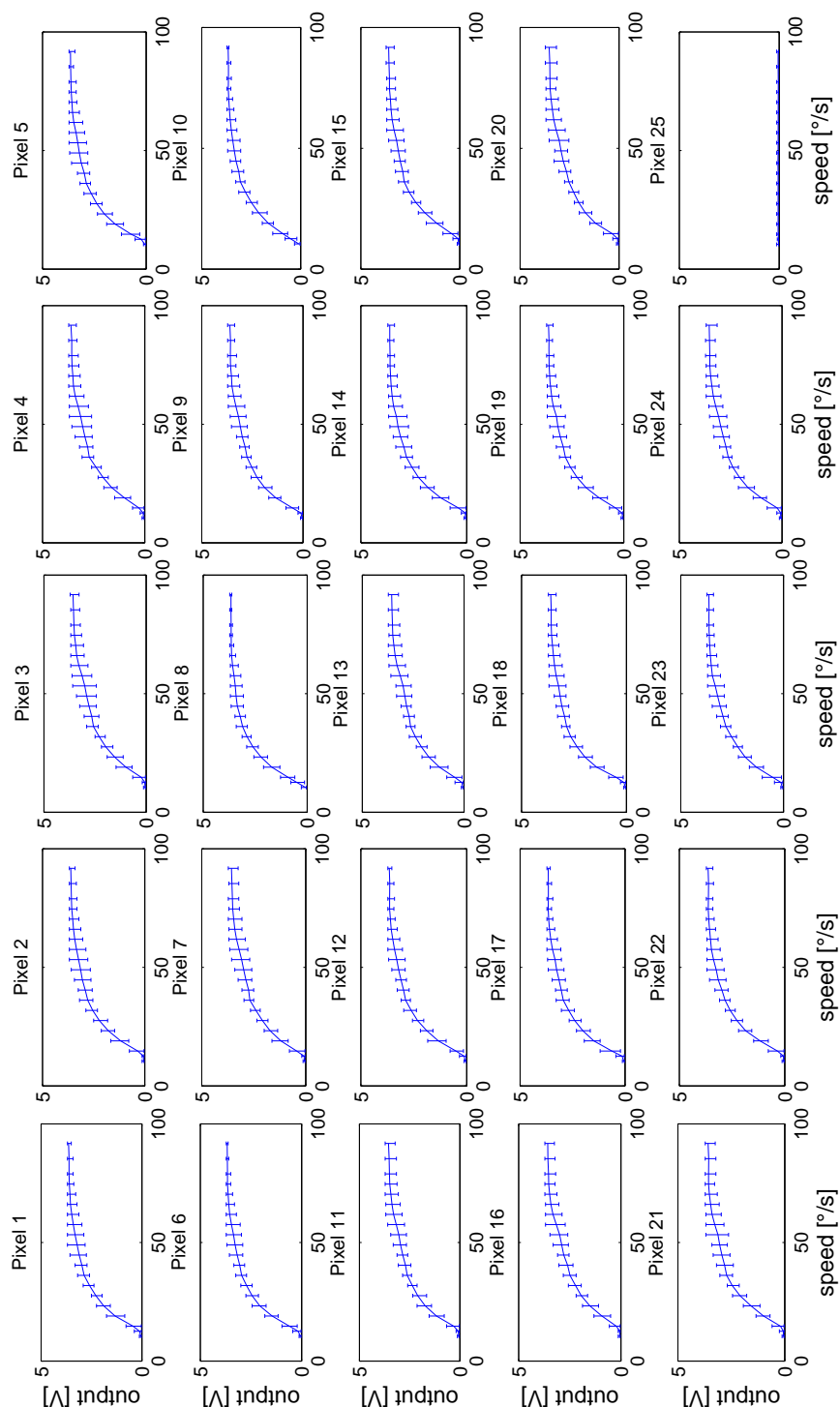


Figure 7.24: Optic flow outputs from the circuits that are sensitive for stimuli moving from the right to the left side of the sensor's pixel array. Outputs have been recorded from all the 24 individual optical flow pixels. Lines are showing mean response averaged over 1000 trials. Error bars indicate maximum and minimum deviation from the mean. Pixel 25 is the sensor's output when no pixel is selected by the scanner.

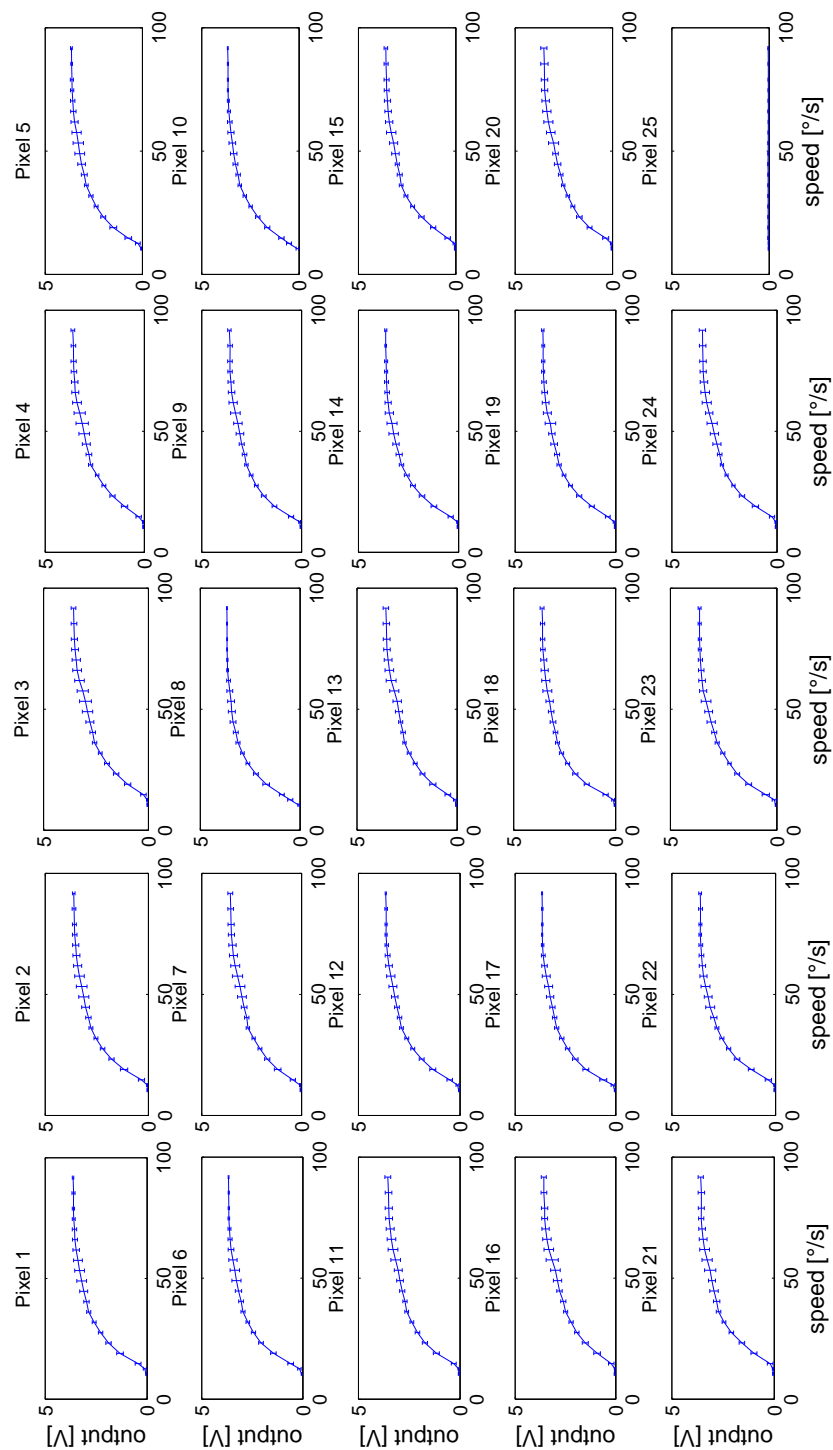


Figure 7.25: Optic flow outputs from the circuits that are sensitive for stimuli moving from the right to the left side of the sensor's pixel array. Outputs have been recorded from all the 24 individual optical flow pixels. Lines are showing mean response averaged over 1000 trials. Error bars indicate the standard deviation from the mean. Pixel 25 is the sensor's output when no pixel is selected by the scanner.

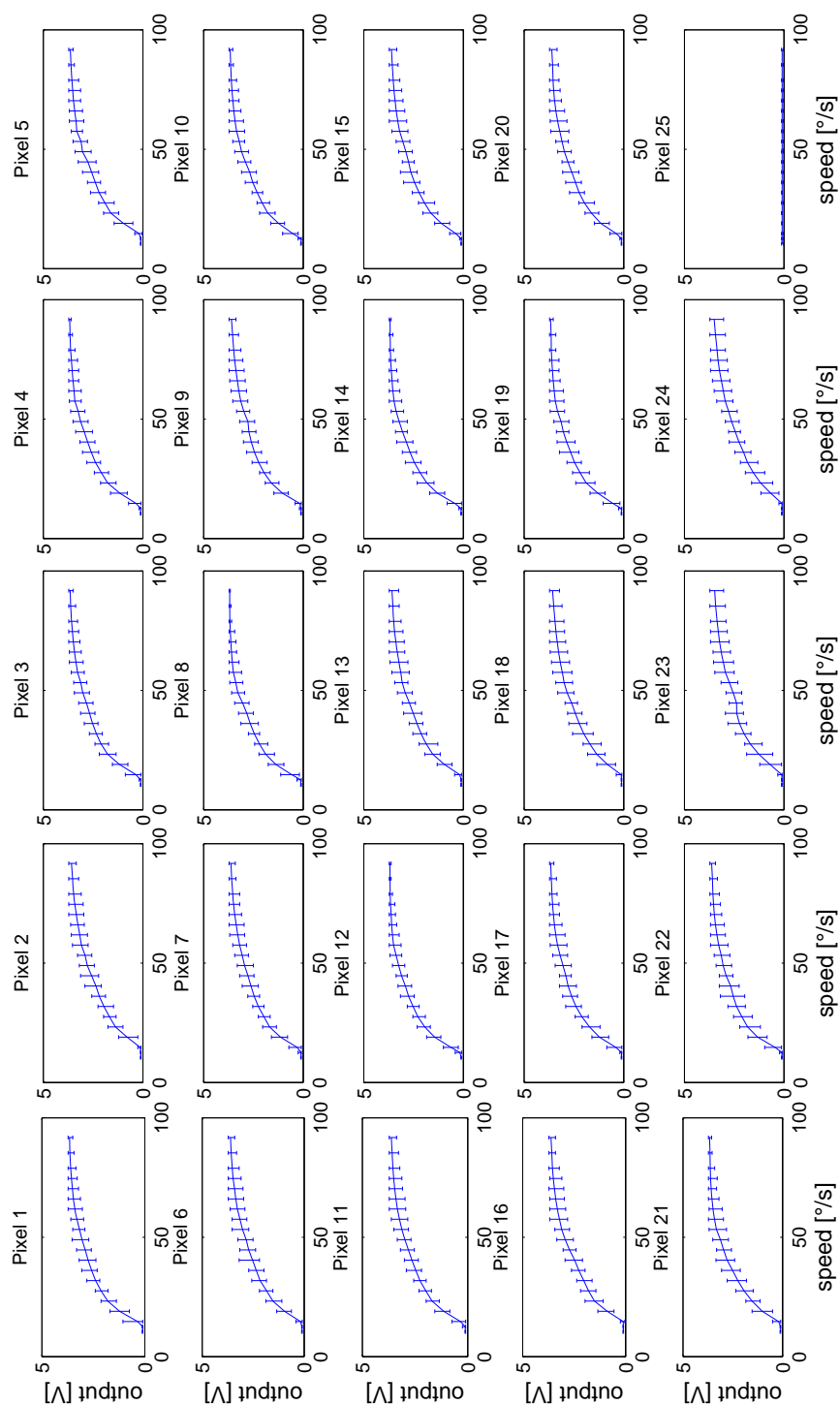


Figure 7.26: Optic flow outputs from the circuits that are sensitive for stimuli moving from the left to the right side of the sensor's pixel array. Outputs have been recorded from all the 24 individual optical flow pixels. Lines are showing mean response averaged over 1000 trials. Error bars indicate maximum and minimum deviation from the mean. Pixel 25 is the sensor's output when no pixel is selected by the scanner.

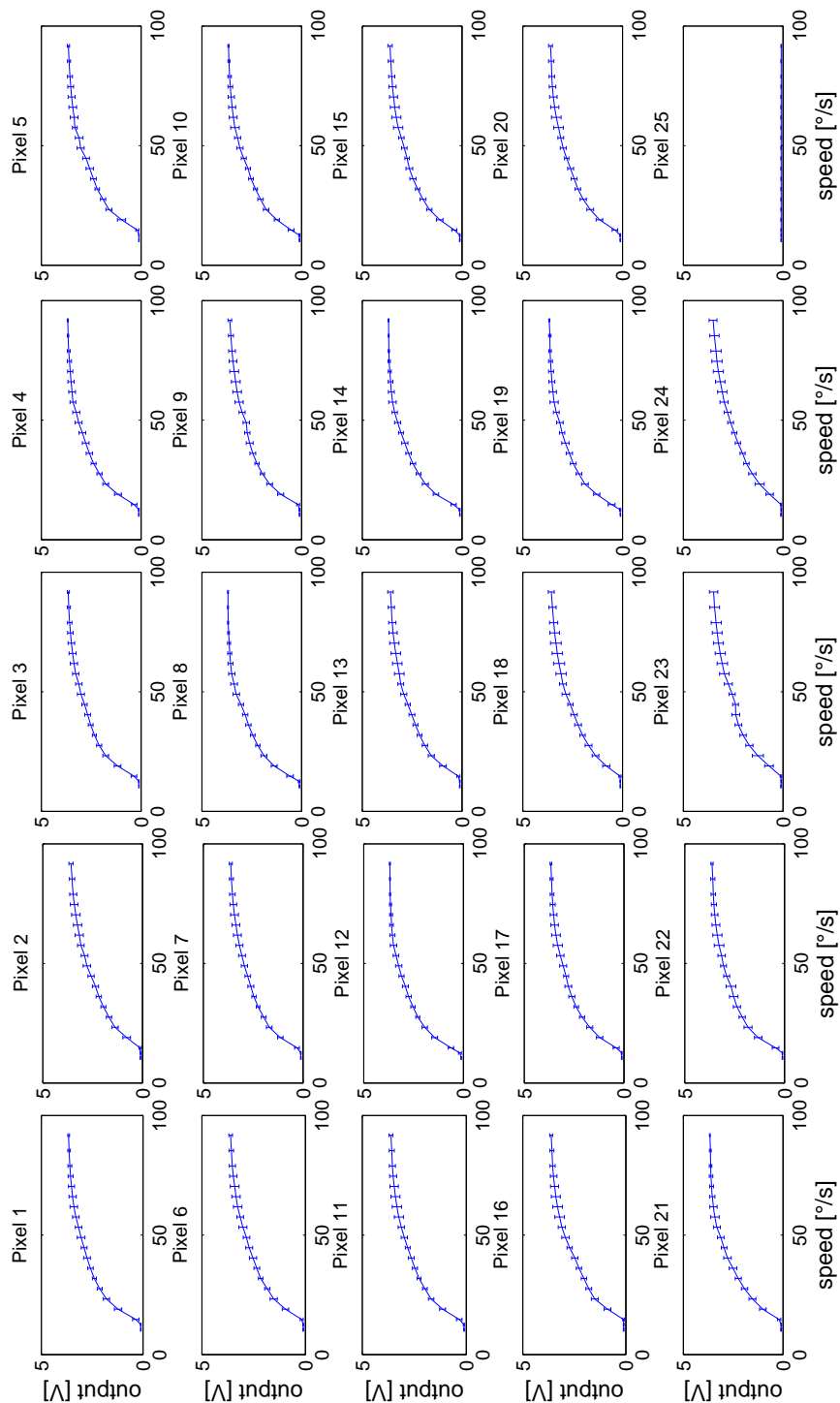


Figure 7.27: Optic flow outputs from the circuits that are sensitive for stimuli moving from the left to the right side of the sensor's pixel array. Outputs have been recorded from all the 24 individual optical flow pixels. Lines are showing mean response averaged over 1000 trials. Error bars indicate the standard deviation from the mean. Pixel 25 is the sensor's output when no pixel is selected by the scanner.

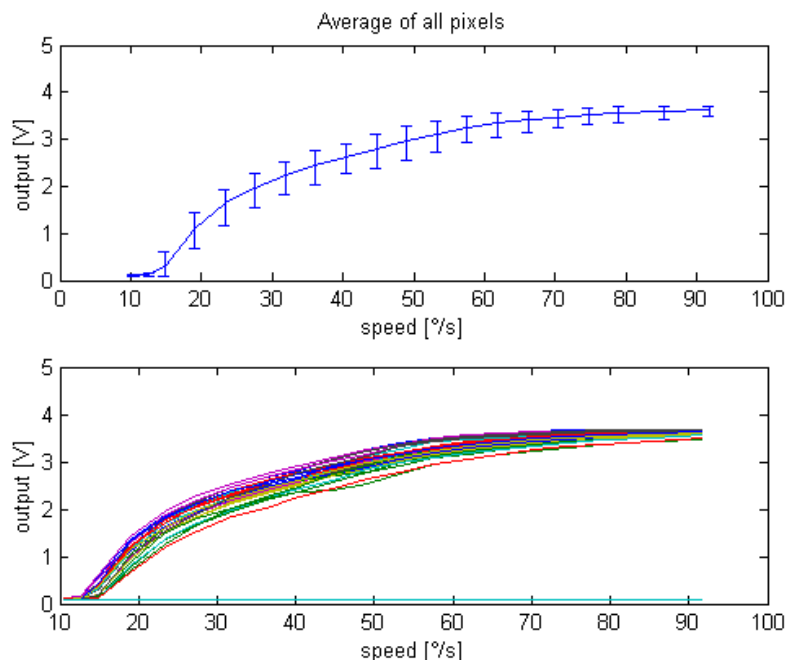


Figure 7.28: Top: Optic flow outputs from the circuits that are selective for stimuli moving from the right to the left. Outputs are averaged across all 24 pixels. Error bars indicate minimum and maximum deviation. Bottom: individual mean optical flow values from the 24 pixels and the sensor's output when no pixel is selected by the scanner. The deviation between pixels outputs shows the effect of fabrication mismatch. Figures adapted from Moeckel and Liu (2007).

7.3.4 Contrast independence of optical flow output

To steer MAVs reliably in natural environments, the on-board vision sensors should be sensitive to low contrast stimuli. The DOFS satisfies this criteria, as the motion output is still computed correctly even for stimuli of contrast values between 2.5% and 100%. The results of this experiment are shown in figure 7.36. The optical flow output stays constant over the whole contrast range for a constant stimulus speed of $35^\circ/s$. In contrast to the optical flow output, the peak-to-peak amplitude of the photoreceptor output (PR) decreases with decreasing stimulus contrast as expected. The results also show that although the individual measured motion output varies across the motion pixels because of the fabrication mismatch, each pixel shows a constant motion output over the specified contrast range.

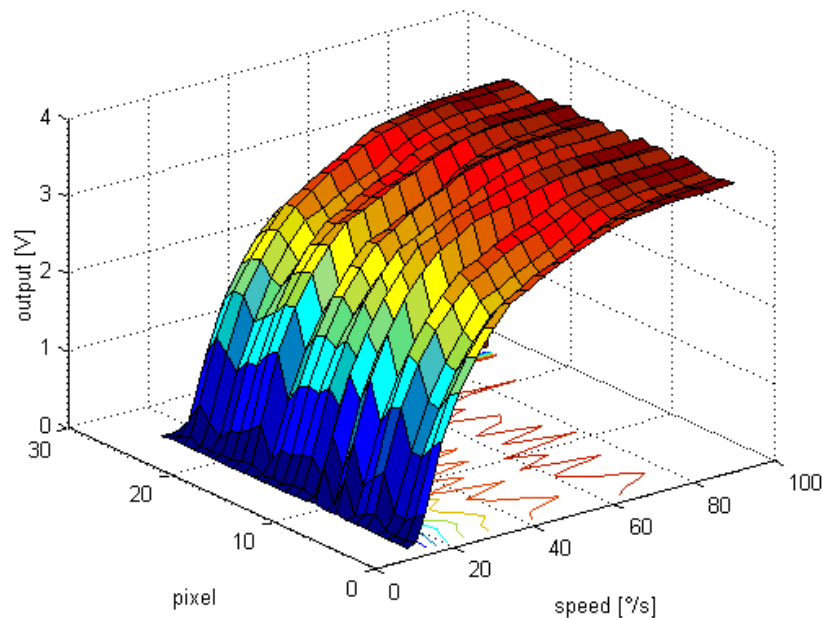


Figure 7.29: 3D plot of the mean optical flow outputs from the circuits that are selective for stimuli moving from the right to the left side of the sensor's pixel array. Characteristic curves are organized according to the pixel order in the sensor array. The sinusoidal pattern across pixels is typical for fabrication mismatch due to striation effect.

7.3.5 Perception of self-rotation

To compare the outputs of the DOFS chip to the output of a gyroscope, we operated the chip on a rotating robot. The results of this experiment have been published in Moeckel and Liu (2008).

After the characterization of the optical flow output of individual pixels a look-up table was computed that allows a correction for the non-linear relation between optical flow output and stimulus speed. This look-up table was used to correct the averaged optical flow output and allowed us to compare the output of the DOFS chip to the output of a gyroscope while accelerating and decelerating the robot platform. The results are shown in figure 7.37. The platform was accelerated and decelerated by approximately $200^\circ/s^2$ with a period of $1.9s$ (figure 7.37a) and by approximately $100^\circ/s^2$ with a period of $0.8s$ (figure 7.37b). We used a gyroscope ADXRS150 from Analog Devices. The gyroscope provided an approximately linear rate output for a maximum yaw rate of $150^\circ/s$. We manually removed the offset of the gyroscope in the plot. The optical flow outputs as well as the output of the gyroscope were sampled with a rate of $100\text{Samples}/s$. To

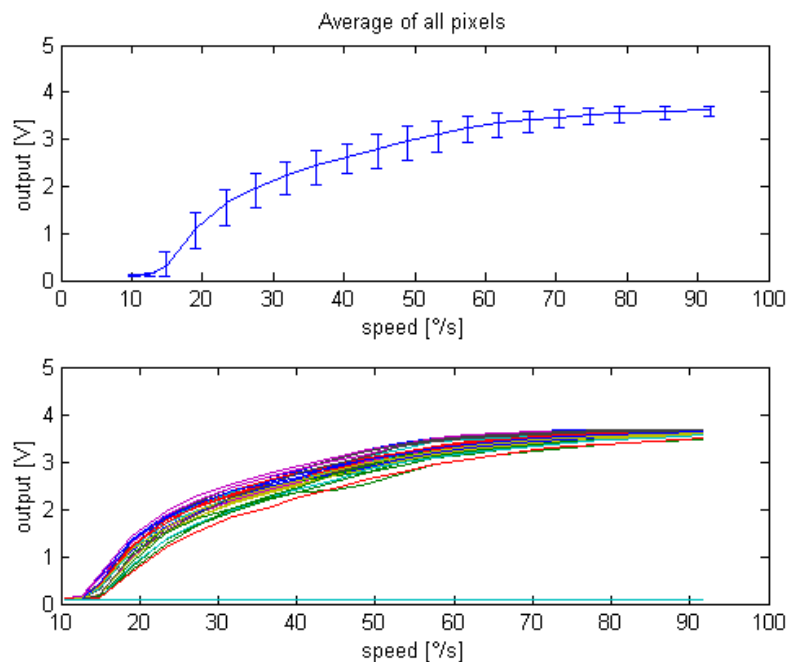


Figure 7.30: Top: Optic flow outputs from the circuits that are selective for stimuli moving from the left to the right. Outputs are across all 24 pixels. Error bars indicate minimum and maximum deviation. Bottom: individual mean optical flow values from the 24 pixels and the sensor's output when no pixel is selected by the scanner. The deviation between pixels outputs shows the effect of fabrication mismatch. Figures adapted from Moeckel and Liu (2007).

remove the temporal noise from the optical flow outputs that were caused by the step motors of the robot, we low-pass filtered the outputs from both the DOFS chip and the gyroscope. The filtered optical flow outputs from the gyroscope and DOFS chip match quite well with a maximum error of $10^\circ/s$. These errors mostly occurred at the beginning and at the end of an acceleration and deceleration cycle and are due to the fact that the platform was not fully damped which led to platform vibrations. The good matching between the outputs of both sensors should allow us to remove the rotational optical flow by subtracting the output of the gyroscope from the overall optical flow output of the DOFS. Thus the output of the DOFS is suitable for control strategy used in state-of-the-art optical flow controlled MAV where the distance dependent translational optical flow component is extracted by subtracting the output of a gyroscope from the total optical flow measured by an optical flow sensor (chapter 4).

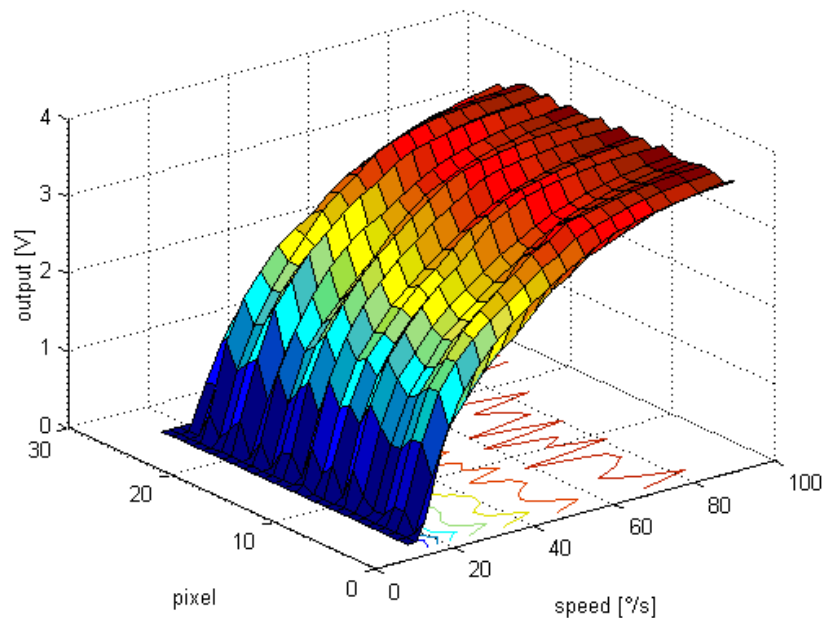


Figure 7.31: 3D plot of the mean optical flow outputs from the circuits that are selective for stimuli moving from the left to the right side of the sensor's pixel array. Characteristic curves are organized according to the pixel order in the sensor array. The sinusoidal pattern across pixels is typical for fabrication mismatch due to striation effect. Figures adapted from Moeckel and Liu (2009).

7.4 Discussion

This chapter describes a dynamic optical flow sensor (DOFS) design that implements a facilitate-and-sample algorithm for optical flow estimation in analog VLSI technology. At the core of this sensor is a novel double threshold edge detection circuit that allows for robust detection of temporal contrast edges. In combination with the photoreceptor circuit by Delbrück and Mead (1994, 1995) and the modified version of the LMC circuit by Liu (2000), the edge detection circuit forms a robust and reliable front-end that can be of benefit for token-based vision sensors in general. The front-end photoreceptor adapts over 4 decades of background intensity and optical flow information can be extracted for visual stimuli with contrast above 2.5%. Stimulus velocities varying over two orders of magnitude can be extracted. The chapter also presents characterization experiments of both of the sensor itself and in combination with a rate gyroscope to illustrate its use for autonomous control of MAVs. These experiments show good matching between both sensors with a maximum error of $10^\circ/\text{s}$.

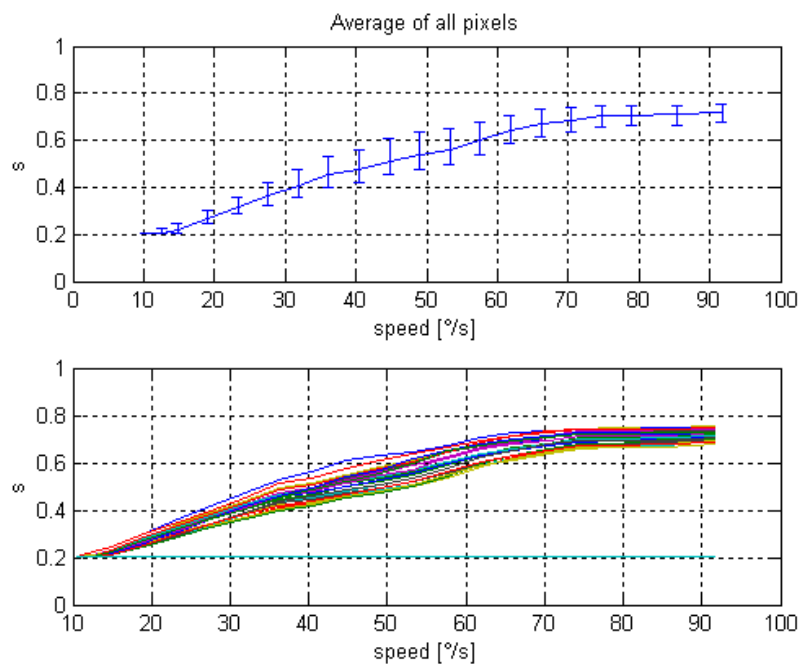


Figure 7.32: Optic flow outputs from figure 7.28 replotted to show the linear theoretically predicted linear relationship between $s = 1/(VDD - output)$ and stimulus velocity.

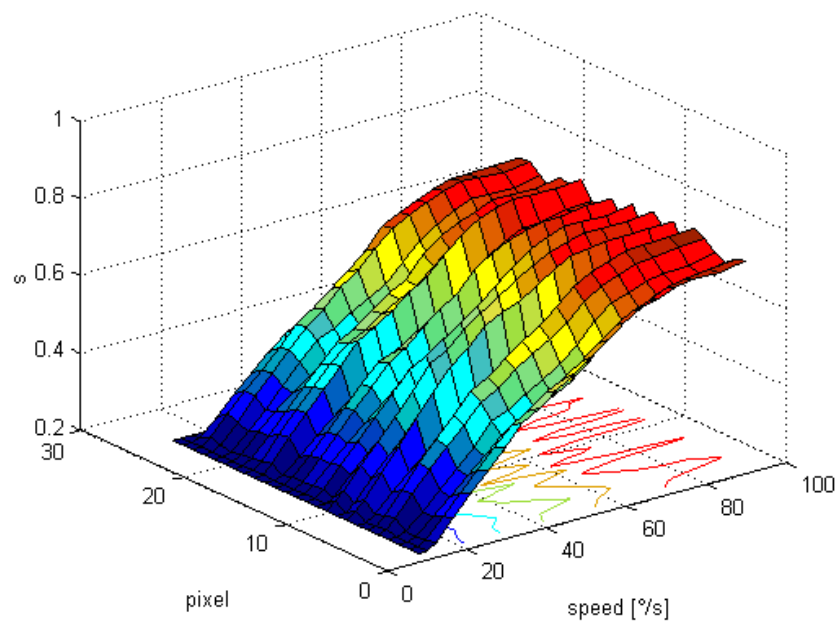


Figure 7.33: Optic flow outputs from figure 7.29 replotted to show the linear theoretically predicted linear relationship between $s = 1/(VDD - output)$ and stimulus velocity.

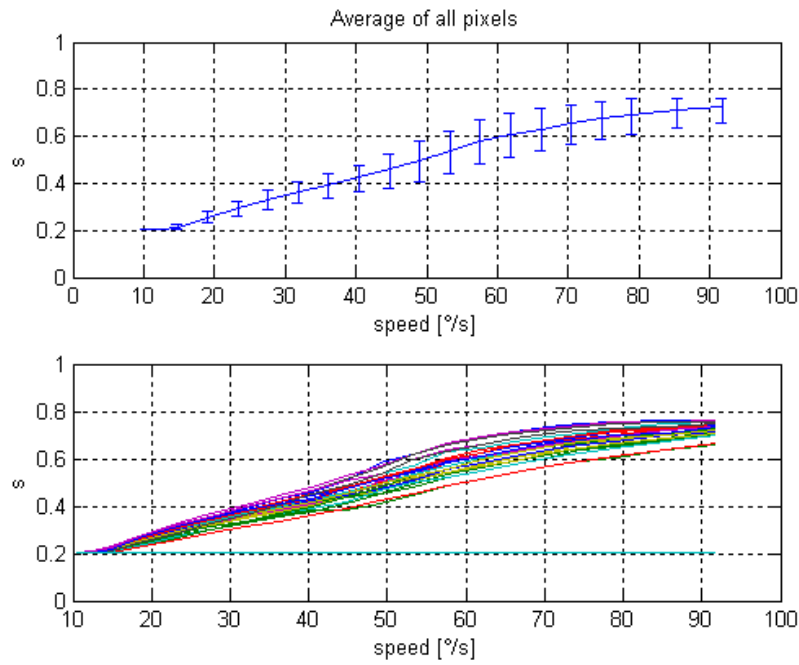


Figure 7.34: Optic flow outputs from figure 7.30 replotted to show the linear theoretically predicted linear relationship between $s = 1/(VDD - output)$ and stimulus velocity.

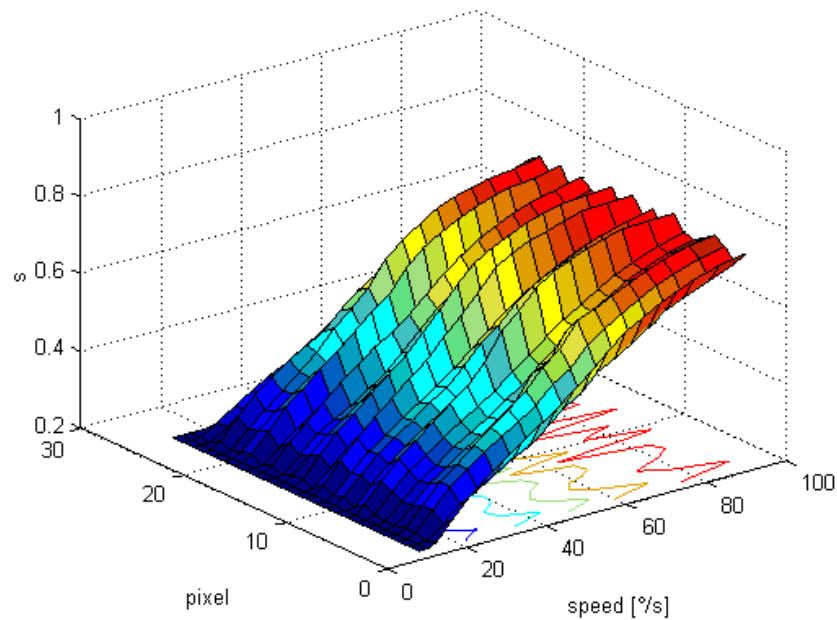


Figure 7.35: Optic flow outputs from figure 7.31 replotted to show the linear theoretically predicted linear relationship between $s = 1/(VDD - output)$ and stimulus velocity.

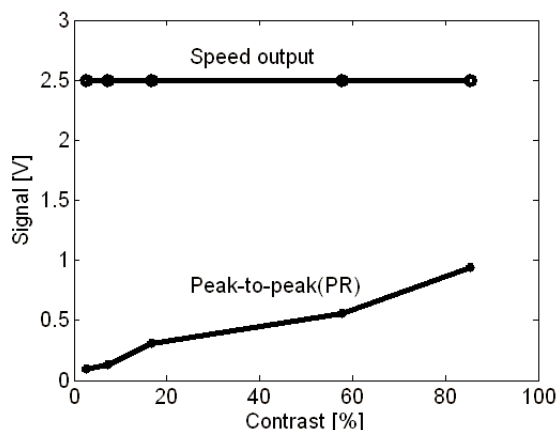


Figure 7.36: Average motion output O and photoreceptor output (PR) of the chip in response to a sinusoidal pattern whose contrast C ranges from 2.5% to 80%. The contrast $C = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} * 100\%$ where I_{max} and I_{min} are the maximum and minimum light intensities of the stimulus measured with a light meter. The range of contrasts is limited by the passive LCD screen on which the pattern is displayed. The distance between chip and screen was approximately 16cm. The motion output stays constant for a constant stimulus velocity even for stimulus contrasts as low as 2.5% while the peak-to-peak amplitude of the photoreceptor (PR) output drops as the stimulus contrast decreases. Figures adapted from Moeckel and Liu (2009).

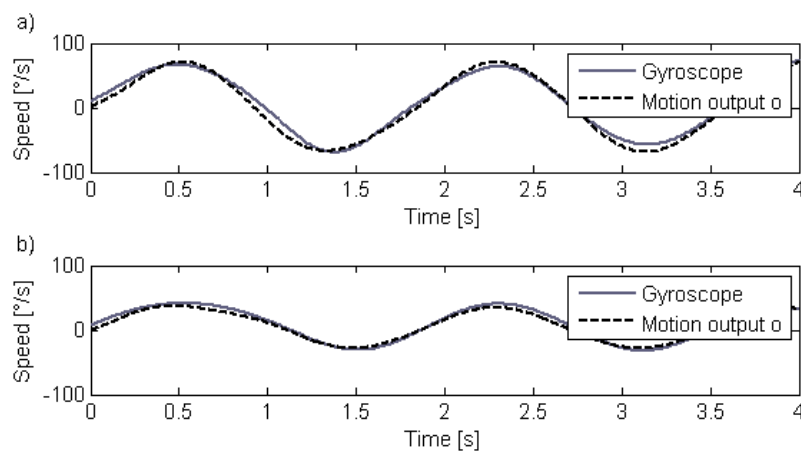


Figure 7.37: Comparison of the optical flow output (dashed black line) with the output of a gyroscope (solid gray line) for two different accelerations ((a) $200^\circ/s^2$, (b) $100^\circ/s^2$). The measurements were taken while the chip and the gyroscope were operated together on a rotating robot. The optical flow output was averaged across all 24 motion pixels, corrected to be linear with speed and low-pass filtered. The maximum error is $10^\circ/s$. Figures adapted from Moeckel and Liu (2008).

Some potential of the FS algorithm remains unexplored in the present dynamic optical flow sensor implementation:

Reducing the power consumption of DOFS

In theory the FS algorithm allows for **very low power** implementations. The analog front-end can be implemented using circuits operating in subthreshold regime consuming little power and the power of the event-based FS circuits can be switched off while no contrast edge is processed. In practice however, the present DOFS implementation is not very low power but consumes several mW of power. This is because of limitations due to the fabrication process several biases (circuit parameters) had to be shared between circuits which does not allow optimal parameter tuning for all circuits. So several buffer circuits that share the same biases have a power consumption that is too high. Also the pulse-shaping circuit that was copied from an earlier unpublished FS design consumes too much power. Furthermore, circuits that switch off power when no contrast edge is detected could have been used. However, I found their implementations too risky for the first design.

Figure 7.38 depicts a modified version of the original adaptive logarithmic wide-dynamic-range photoreceptor by Delbrück and Mead (1994, 1995) that would allow to reduce also the power consumption of the DOFS photoreceptor circuits. Although the present photoreceptor circuit (section 7.2.1) can be operated in subthreshold it is not exactly a low power circuit. This is because the current in the inverting amplifier - controlled by the bias, V_{prbias} - has to be large enough to make the feedback loop sufficiently stable at high illumination levels. So if V_{prbias} is set statically the photoreceptors current consumption has to be adjusted for the highest expected photocurrents and the photoreceptor constantly burns several μA although on average a bias current through M3 in the order of nA would be sufficient to keep the feedback loop stable.

Delbruck and Oberhoff (2004) proposed a modification of the original photoreceptor circuit where V_{prbias} and thus the current in the inverting amplifier is automatically adjusted according to the average photocurrent. Figure 7.38 shows the schematics of this low-power self-biasing photoreceptor circuit that for example has been successfully demonstrated in the vision sensors designs by Lichtsteiner et al. (2006, 2008). In this modified PR design the photocurrents of all sensor pixels are sourced through the diode-connected MOSFET M0 that copies a fraction of the photocurrent into the individual pixels by the MOSFET M3. The size ratio of M0 and M3 sets the precise ratio of currents. The capacitor C0 has to be large enough to avoid unstable behavior in

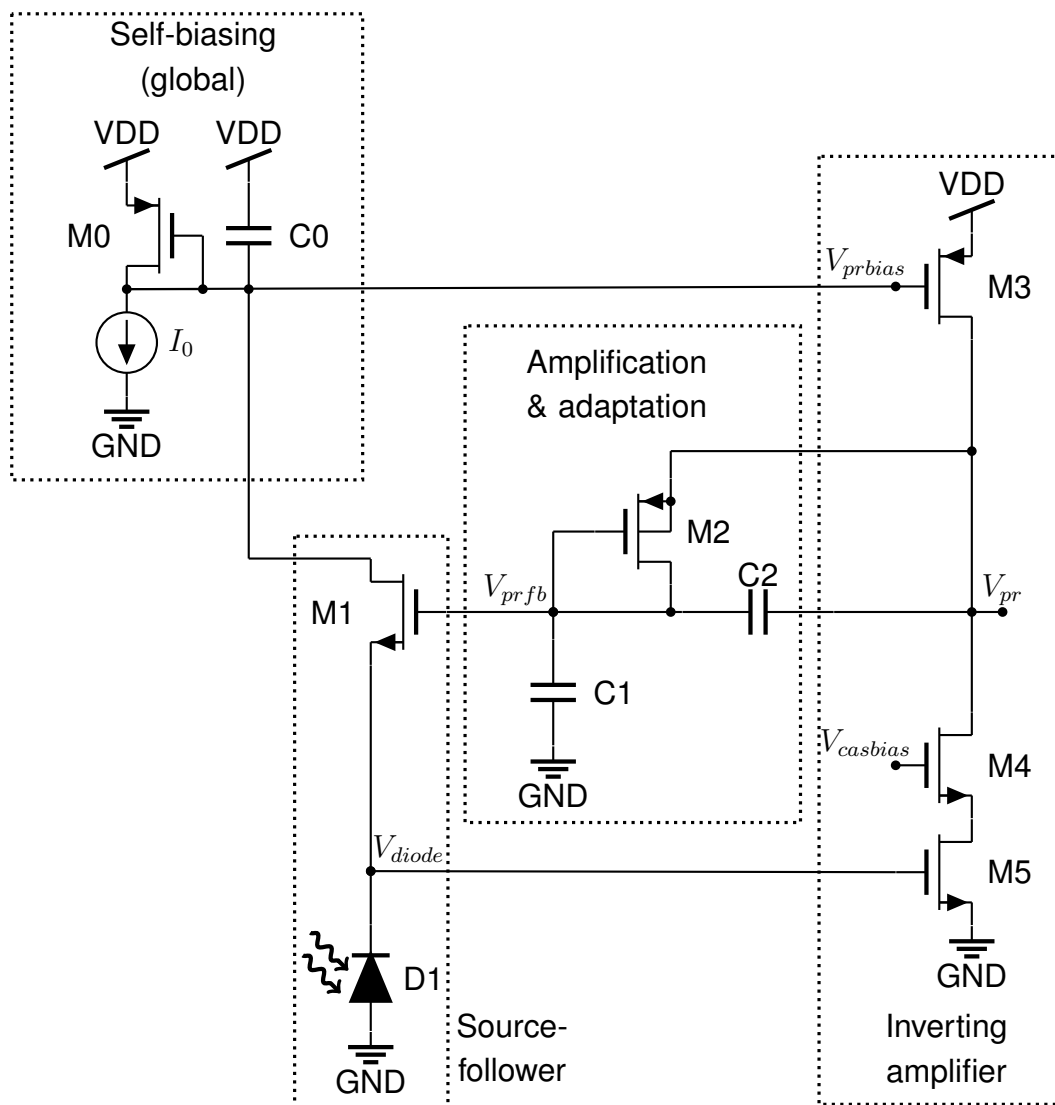


Figure 7.38: Self-biasing photoreceptor circuit.

the feedback loop that adjusts V_{prbias} . The current source I_0 generates a small offset current that ensures that at low photocurrents the inverting amplifiers stay turned on. M_0 , C_0 and I_0 have only to be implemented once per pixel array. So the overall pixel size remains constant.

The self-biasing adaptive photoreceptor by Delbruck and Oberhoff (2004) inherits all the properties of the original adaptive photoreceptor by Delbrück and Mead (1994, 1995) (section 7.2.1) including its amplification of temporal contrast.

An event-based communication interface for DOFS

The non-continuous event-based optical flow computation of the FS algorithm opens the possibility to implement event-based communication between the sensor and a host processor that is running the controller for an autonomous robot. Similarly as the retina design by Lichtsteiner et al. (2006, 2008) the individual pixels of a FS optical flow sensor could communicate with the host processor when a new optical flow estimate is available using an interrupt-based communication scheme. Communication schemes that use the interrupt of a microcontroller can be very efficient especially for sparse communication when not every optical flow pixel detects a contrast edge in the visual stimuli.

In the present DOFS design, such an event-based communication scheme remains unexplored. Instead I am sequentially scanning the output of all optical flow pixels. The drawback of this scanner scheme is that the analog outputs of the DOFS need to be digitized leading to a less efficient communication, higher computational load for the host processor and higher load for the analog-to-digital converter. The advantage lies in the simplicity of the communication that makes the system very robust. Furthermore, the technique of constant scanning the optical flow outputs corresponds better to the typical frame-based computation as it is still standard in many control and machine vision applications - allowing for a direct integration of DOFS into existing control architectures. The continuous readout of the analog optical flow outputs allows for an efficient coding of both the optical flow value as well as the point in time when the optical flow value was updated in the same analog signal. This way the continuous analog optical flow output can carry both the optical flow information as well as a reliability measure in the same signal.

In future designs, the full potential of the event-based optical flow computation in the DOFS design could probably be best explored by adding an event-based communication interface to the sensor. Asynchronous communication interfaces with address-event representation (AER) have been widely studied for instance by Boahen (2000, 2004a,b,c). An AER interface allows individual pixels in an array to output a digital event by placing their address on a global data bus. Facilitate-and-sample sensor pixels could benefit from such an interface in the following way. Individual pixels could trigger an address event only when a new optical flow measurement is available. The precise optical flow value remains stored on a capacitor until the value is read by a host device like a microcontroller and the event is acknowledged. The combination of the asynchronous event-based optical flow circuits together with an asynchronous event-based AER communication interface allows the microcontroller to only read the outputs

of those optical flow pixels that provide a new measurement thus saving communication time and reducing both the power consumption of the overall system as well as the computational load on the host controller. The microcontroller and its internal analog-to-digital converter do not have to continuously read, convert and process all optical flow output signals on every clock cycle but only once per optical flow value update by the FS circuits.

Facilitate-and-sample circuits for a linearized optical flow output

The facilitate-and-sample (FS) algorithm measures the time t that a contrast edge requires to travel the distance between neighboring pixels. As a result, a FS optical flow sensor which uses a linear decay rate for its facilitate signal, will generate an output OF that is not linearly proportional to the speed v of the stimuli traveling across the sensor's focal plane but to t . As discussed in section 7.1.2, this does not have to be a disadvantage. If, however, a linear or more complex $OF-v$ relationship is desired, this can be achieved by using a piece-wise approximation of the preferred output function.

Figure 7.39 shows the design of a new circuit that can generate linear and complex $OF-v$ output functions using piece-wise approximations. The circuit uses the capacitor C5 that stores the facilitate signal F as well as the reset MOSFET M55 and the current source M56 from the original facilitate circuit in figure 7.15a. It includes in addition, several comparators (OTA1-OTAn) that deactivate the current paths through additional current sources (M57, M59, M61) when the F -signal falls below each OTA's corresponding threshold voltage ($V_{thr1bias}-V_{thrnbias}$). So as F decreases over time, the number of active current sources decrease, and the decay rate $\Delta F/\Delta t$ decreases as well.

The simulation plots in figure 7.40 show optical flow OF output profiles that are linear in the stimulus velocities v (figures 7.40b and 7.40d). They also show the corresponding timing diagrams for the facilitate signals (figures 7.40a and 7.40c) using 7 and 4 different thresholds and additional decay rates, respectively. Horizontal lines in the plots depict the thresholds that are used for linearization. For generating the linear output profiles in (b) and (d), I use the pulse width of the nP -signal (dotted line in figures 7.40a and 7.40c) that resets F , to limit the maximum stimulus velocities that can be resolved. With the various threshold and decay rate parameters, the circuit using 7 (4) thresholds generates an RMS error of 1.68% (5.48%) and a maximum error of 3.03% (14.90%) in comparison to a perfect linear relationship.

The circuits using several thresholds and decay parameters will benefit from a dynamic parameter update that adjusts the range of stimulus velocities that can be optimally

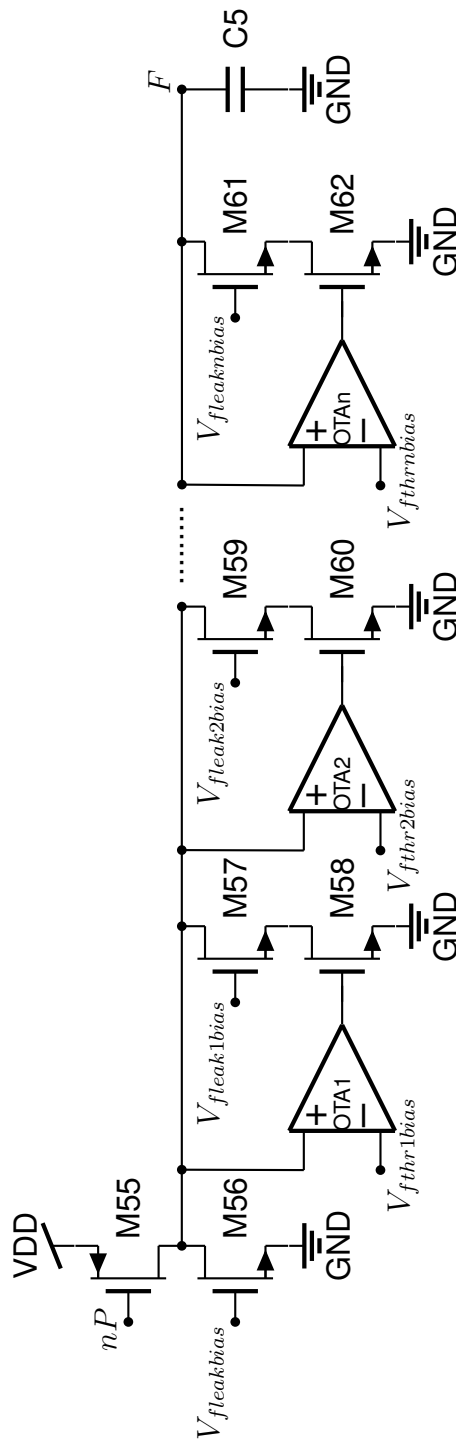


Figure 7.39: Multi-decay rate facilitator circuit which produces an output that is approximately linear in the stimulus speed.

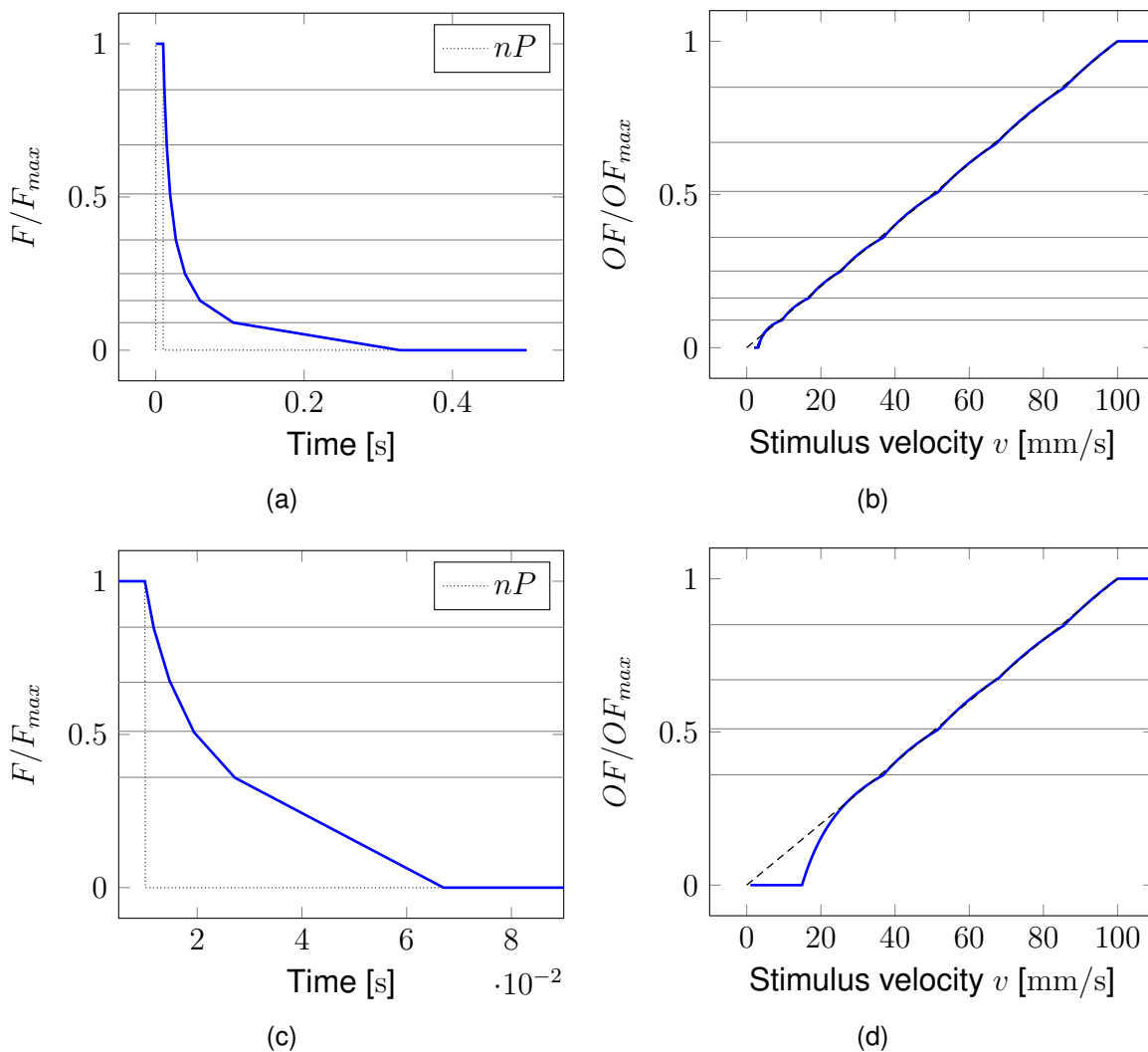


Figure 7.40: Simulation plots of the multi-decay rate facilitate circuit. (a,c) F vs time (b,d) F vs stimulus velocity. (a,b) 7 thresholds rms error:1.68%; max error:3.03% (c,d) 4 thresholds rms error=5.48%; max error=14.90%

measured and resolved similarly as the fabricated DOFS.

To the best of my knowledge such a multi-threshold facilitate circuit that supports an optical flow output that changes linearly with stimulus velocity has not been demonstrated in the literature before.

Chapter 8

Steering a simulated car with the dynamic optical flow sensor

Abstract

To demonstrate the capabilities and potentials of the dynamic optical flow sensor (DOFS) a simplified car driving simulation was developed in Matlab. DOFS observes the simulation through a standard LCD laptop screen. We trained a single-layer perceptron - a well-known numerical neuron model - to process the local optical flow outputs generated by DOFS to steer the car in the driving simulation. The presented work has been published at the IEEE International Symposium on Circuits and Systems (Moeckel et al., 2008). I would like to thank Roger Jäggi who developed with me the car driving simulation as part of his semester project at the Institute of Neuroninformatics.

The following section 8.1 presents the car driving simulation software and experimental setup while section 8.2 describes the perceptron controller. Section 8.3 presents experimental results where DOFS is demonstrated to autonomously steer the simulated car based on visual feedback. The results are discussed in section 8.4.

8.1 Experimental Setup

A picture and the schema of the experimental setup are shown in figure 8.2. The analog local optical flow values (the sampled facilitate signals H_R and H_L) from the 24 DOFS pixels are continuously scanned by an ARM7 microcontroller in a frame-based fashion and converted into digital values with a resolution of 7 bit. The microcontroller also

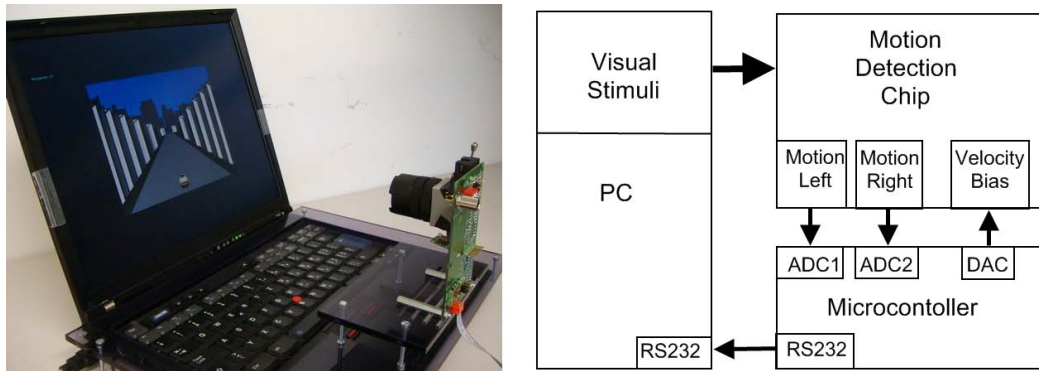


Figure 8.1: Demo setup. The motion chip computes optical flow based on the driving simulation generated by a PC. A microcontroller continuously scans the local motion values from the chip, converts the analog values to digital ones and sends them via a serial communication interface (RS232) to the PC where a simple controller updates the position of the simulated car based on the motion values. The microcontroller is further responsible for adjusting the velocity bias that sets the current detectable range of image velocities. Figures adapted from Moeckel et al. (2008).

adjusts the velocity bias, $V_{leakbias}$, that sets the range of image velocities that DOFS can detect (see section 7.2.4).

During the experiments the microcontroller's analog-to-digital converter was set to its maximum sampling rate of 10KSamples/s which leads to a minimum readout time of $100\mu\text{s}$ per pixel. But since the optical flow measurements change only slowly over time we read the sensor outputs not at the maximum frame rate but at a reduced rate of 100 frames per second. The optical flow values were sent to a laptop PC running the driving simulation as well as the controller of the simulated car via a standard serial RS232 communication interface with a baud rate of 38400baud/s .

The car driving simulation and controller are implemented in Matlab. The visual stimuli are generated with the Psychtoolbox (Brainard, 1997; Pelli, 1997). A typical screen shot of the game is shown on the left side of figure 8.2. During the game the car at the bottom of the screen is steered using the local optical flow outputs generated by DOFS to avoid collisions with the walls on both sides of the simulated road. The sensor was placed so that it monitors the far end of the street as indicated by the red box. For a straight street, the optical flow sensor outputs a 1-dimensional optical flow field that forms an expansion pattern similarly to the one shown on the right of figure 8.2. If the street curves to the left or right of the screen, the optical flow field is shifted and this shift is used to adjust the position of the car in the next frame.

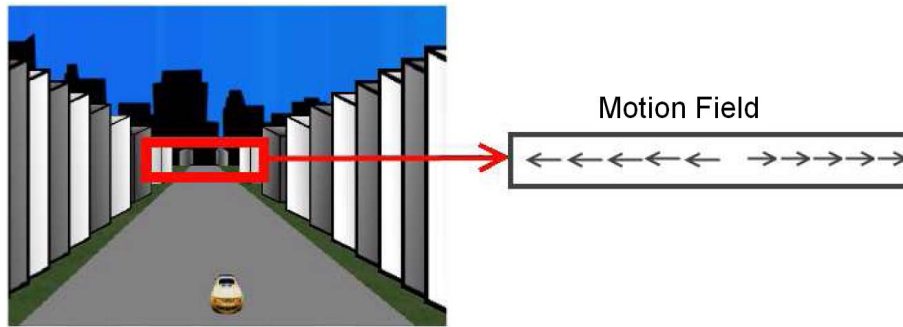


Figure 8.2: Screen shot of one frame in the car driving simulation. The visual field of the motion detection chip is indicated by the red box. The box on the right shows the local motion values of the chip that are used to steer the car. Figure adapted from Moeckel et al. (2008).

8.2 Single-layer perceptron controller

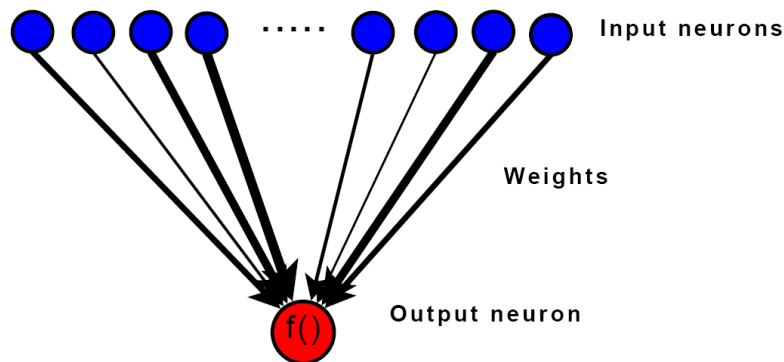


Figure 8.3: The perceptron calculates the current position of the street based on the 24 DOFS optical flow output signals. Figure adapted from Moeckel et al. (2008)

For steering the simulated car we decided on a simple controller based on the well-known perceptron shown in figure 8.3. The 24 optical flow values extracted by DOFS are inputs to the perceptron that fuses the information from the individual optical flow pixels and produces an output based on the weighted sum of the inputs following the equation:

$$output = f\left(\sum_{i=1}^{24} weight_i * input_i\right) \quad (8.1)$$

where f is the sigmoid function. During a training period the weight vector is updated

using the following supervised learning rule

$$\Delta weight = \eta * (supervisor - output) * input \quad (8.2)$$

where the weight change $\Delta weight$ is determined based on the difference between the *supervisor* signal (that is the desired output) and the current *output* of the perceptron. η is a constant that is used to adjust the amount of weight update per learning step.

8.3 Experimental results

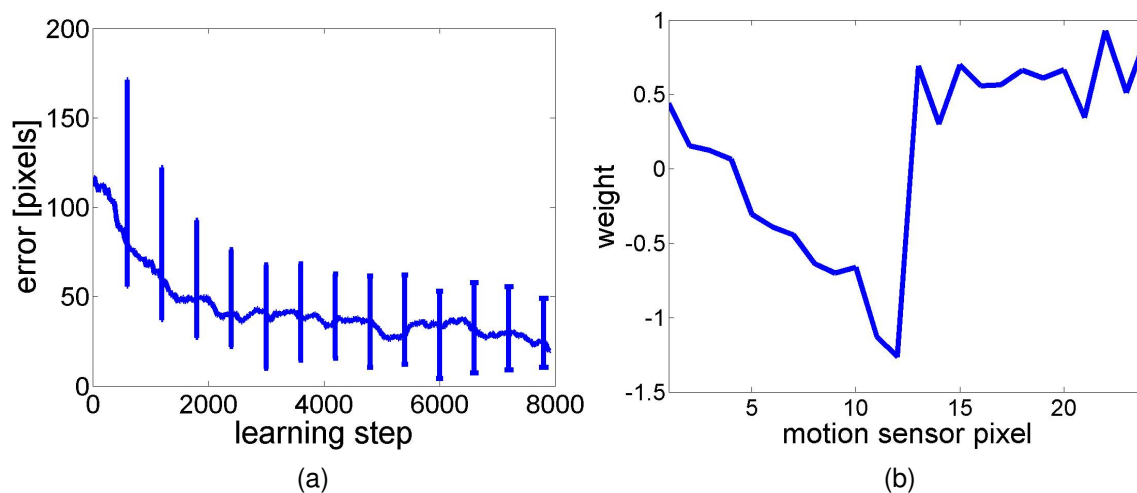


Figure 8.4: (a) Average estimation error of the current position of the center of the street measured in number of pixels on the screen. Error bars indicate the standard deviation. After the supervised training phase of the perceptron, the error decreases below 60 pixels. (b) Final weight values for the inputs of the perceptron after 8000 learning steps. The perceptron learned to steer the simulated car towards the center of the street which was approximately in the field of view between optical flow sensor pixels 12 and 13. Figures adapted from Moeckel et al. (2008).

We generated a long driving track for the car simulation. During the training phase, the perceptron was trained with different segments of this track. A typical evolution of the displacement error during this training phase is shown in figure 8.4a. The error value defined in screen pixels is the difference between the distance of the current position of the center of the street and the perceptron output based on the local motion values. The

displacement error decreases below 60 pixels after 8000 weight updates while using a screen resolution of 1024x768 pixels.

An example of a final weight distribution at the end of a learning phase is shown in figure 8.4b. For this experiment, the optical flow sensor was placed in front of the screen so that the center of the street was approximately in the visual field of optical flow pixels in the center of the sensor's 1-dimensional pixel array. Even though the weight distribution is asymmetric, the net output of the perceptron is close to zero which means that the car will stay in its current horizontal position. The unsymmetric weight distribution is probably due to the fact that the motion chip was not looking exactly at the center of the screen or that the sensor was not oriented orthogonally to the screen surface. Overall the perceptron proved to have the potential to correct for both displacement errors of the sensor with respect to the center and orientation of the screen as well as for fixed pattern noise in the optical flow outputs and lens aberrations.

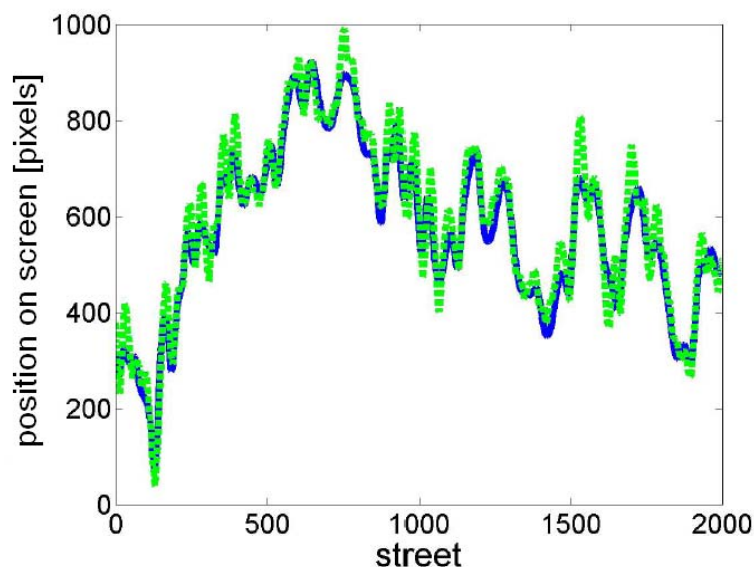


Figure 8.5: Horizontal position of the center of the street (solid line) and the location of the car (dashed line) on the screen in units of pixels. Figures adapted from Moeckel et al. (2008).

The trajectory of the car through the driving simulation after training is shown in form of a dashed line in figure 8.5. The car was able to drive close to the center of the street (depicted as a solid line in figure 8.5) without colliding with the walls. The figure shows that when the street contains sharp curves the car sometimes leaves the middle line of the street - but does not leave the street. This behavior is caused by the proportional steering controller but especially by the limited number of only 24 optical flow pixels that

where not sufficient to cover the entire screen. I believe that a slightly more complex PID controller that allows better estimation of the next street position based on the history as well as the application of a DOFS with a higher spatial resolution could reduce the displacement errors.



Figure 8.6: Screen shots of the car driving simulation with contrast of 1.0 (left) and 0.05 (right). Figures adapted from Moeckel et al. (2008).

We tested the steering properties of our setup for scenes of different contrasts and background light intensity. Examples of two different contrast scenes are shown in figure 8.6. Because of the contrast independent optical flow outputs of DOFS, the setup can deal with low contrast simulations down to a contrast of 5% and various background light intensities.

8.4 Discussion

The presented demonstration setup shows the potentials and properties of the DOFS in steering a simulated car using only 24 optical flow pixels. The car driving simulation is somewhat simplified due to the limited number of pixels in the DOFS and the limited resources for the development of the simulation environment. Future demonstration systems that simulate more complex 3D flight behavior would probably require several DOFSs with a higher pixel count. The present car driving simulation is nevertheless a powerful demonstration system: By fusing the optical flow values with the help of a simple single-layer perceptron, one can find the displacement error of the simulated car in relation to the center of the street and generate a steering signal. The perceptron produces the steering output only based on the weighted sum of the local optical flow values. Because of the inclusion of circuits on the optical flow sensor to model the properties of cells in the stages prior to the optical flow computation in biological systems,

and the learning through the perceptron, we get a robust setup that (i) is adaptive to approximately three orders of magnitude of change in background light intensity; (ii) operates over a large range of contrasts down to a contrast of 5%; (iii) supports pixel-level optical flow variations over two orders of magnitude without bias changes; (iv) adapts to displacement errors of the placement of the motion chip in front of the screen, and (v) is robust to variance across the local measured optical flow values. So the system demonstrates how the difficult task of steering a car can be achieved with the help of the DOFS that performs the computationally expensive task of optical flow processing.

Chapter 9

Adaptive aVLSI vision sensor

Abstract

This chapter presents a 160-pixel adaptive vision sensor (aVIS) implemented in analog Very Large Scale Integrated Technology. By outfitting the vision sensor with a standard lens, a 0.3 gram compact vision system for guidance of micro-aerial vehicles (MAV) was created. The sensor features the wide-dynamic range photoreceptor circuit that adapts over four orders of background light intensity as well as the LMC circuit from the dynamic optical flow sensor presented in chapter 7. Optic flow estimation is implemented on a dsPIC microcontroller that sequentially reads the LMC output signals from aVIS and computes optical flow based on the Image Interpolation Algorithm (I2A) by Srinivasan (1993, 1994). We characterize the system for optical flow computation by comparing the I2A outputs against the outputs from a rate gyroscope and show that the vision system is a suitable candidate for visual guidance of autonomous robots.

Section 9.1 gives an overview of the adaptive vision system module while sections 9.2 and 9.3 introduce the aVIS circuits and image interpolation algorithm, respectively. Experimental results are presented in section 9.4. Section 9.5 concludes and discusses the experimental results.

Highlights and original contributions

This chapter explores the possible advantages of using a front-end bio-inspired sensor with properties such as local background light adaptation over a wide range of intensity levels and local amplification together with an algorithm that computes optical flow based on sets of images. Similarly as the front-end circuits implemented in DOFS (pre-

sented in chapter 7) the photoreceptor and LMC circuit of the adaptive vision sensor amplify temporal contrast of visual stimuli and show adaptation behavior when the stimulus contrast does not vary over time. So since the Image Interpolation Algorithm (I2A) was designed processing optical flow based on consecutively taken images it was not obvious that the I2A could be used as well to compute optical flow from the outputs of the adaptive vision sensor.

9.1 Overview of the adaptive vision system

This section introduces the adaptive vision sensor and explains the history of the sensor.

9.1.1 Sensor overview

Table 9.1: aVIS and lens specification.

Fabrication process	AMS 0.35 μ m 4 Metal 2 Poly
Supply voltage	3.3V
Number of pixels	160 x 1
Pixel distance	20 μ m
Size of photodiode	17 μ m x 66 μ m
Chip size	4.141mm x 2.441mm
Weight incl. PCB	0.117g
Lens name	DSL767A
Focal length	2.43mm
F/#	2.8
Diagonal field of view	66°
Size	8mm x 8mm x 3.2mm
Weight	0.18g

The specifications of the adaptive vision sensor are summarized in table 9.1. A microphotograph of the adaptive vision sensor's bare die is depicted in figure 9.1. The aVLSI sensor consists of a linear array of 160 pixels and was fabricated in an AMS 0.35 μ m 4 metal 2 poly process. Each pixel contains a **photodiode** that converts light into photocurrents. A **photoreceptor circuit** (section 9.2.1) amplifies temporal contrast

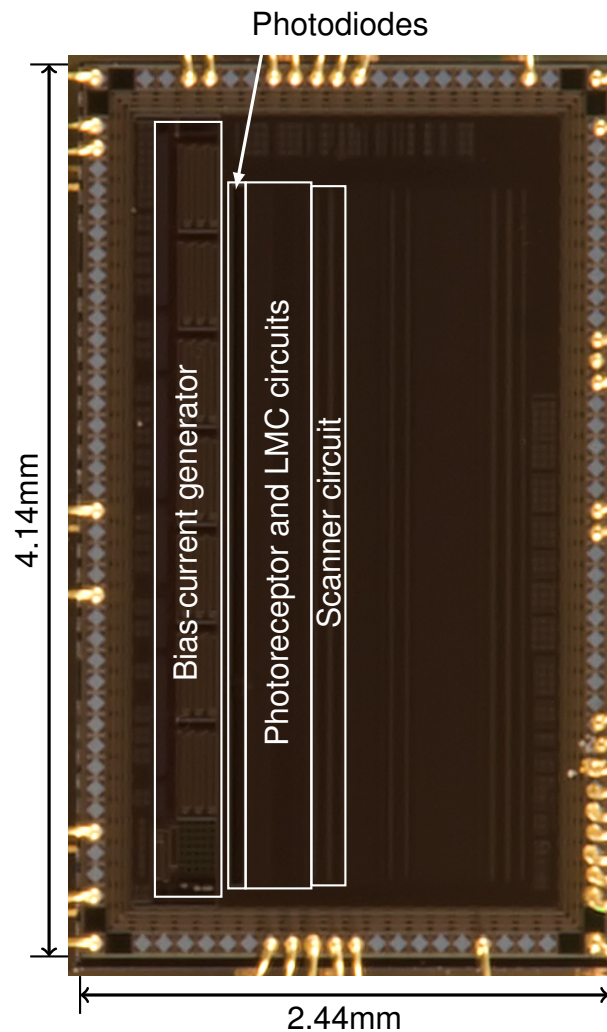


Figure 9.1: Picture of adaptive vision sensor bare die. The vision sensor contains an array of 160 pixels that are composed of a photodiode, photoreceptor and LMC circuit. An analog scanner circuit allows sequential access to local pixel output signals. The bias-current generator provides the capability to generate all bias parameters on-chip. Since aVIS was originally designed as a predecessor of the dynamic optical flow sensor it contains additional circuits. (Picture taken by Adam Klaptocz.)

and adapts to background light intensity (Delbrück and Mead, 1994, 1995). The photoreceptor outputs are further amplified and clamped by an **LMC circuit** (section 9.2.2) that models the Laminar Monopolar Cell (LMC) in the fly visual system (Liu, 2000). The local LMC output signals are sequentially read through an analog **scanner circuit** (section 9.2.3) (Mead and Delbruck, 1991). The sensor furthermore includes a programmable bias generator circuit that allows the on-chip generation of the circuits' bias

currents (Delbruck and Lichtsteiner, 2006). With this circuit block, no external chips like DACs are needed for generating the bias values of the sensor.

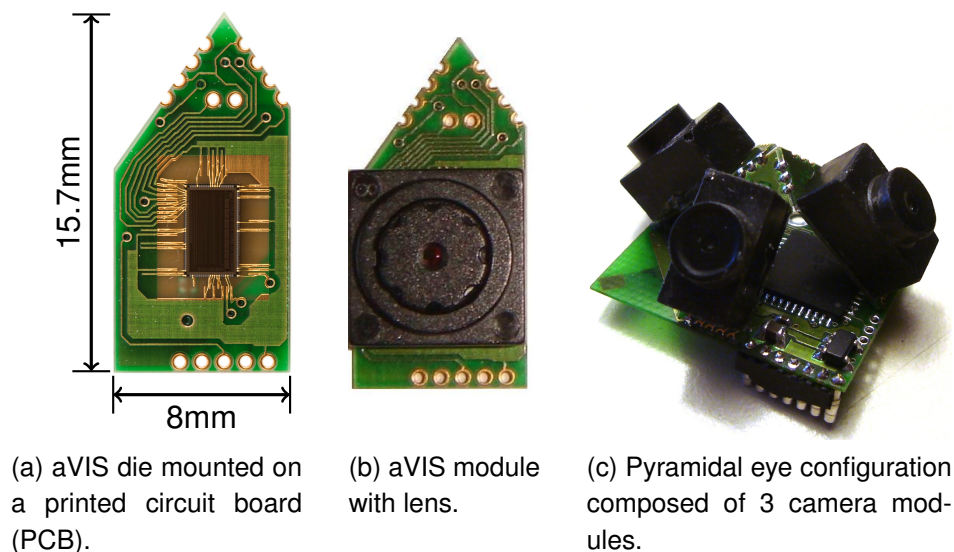


Figure 9.2: (a) aVIS chip die directly bonded on a custom printed circuit board. (Picture taken by Adam Klaptocz.) (b) Adaptive vision sensor module with lens. (c) 3 aVIS modules forming a pyramidal eye. Vision sensors pointing to the left, right, and bottom for obstacle avoidance and altitude control. Design idea by Jean-Christophe Zufferey. The bottom PCB with microcontroller shown in (c) was designed by Adam Klaptocz.

Figures 9.2a and 9.2b show the adaptive vision sensor mounted on a triangle printed circuit board (PCB) without and with lens attached. The specific shape of the PCB was chosen according to an idea by our partner Jean-Christophe Zufferey. The shape allows three to four sensor modules to be mounted together to form a pyramidal eye as depicted in figure 9.2c. The sensor outputs go to a microcontroller mounted on the bottom PCB of the pyramid. The pyramidal configuration was chosen to form a compact vision system where individual sensors - when mounted on the MAV - face to the left and right sides for obstacle avoidance and to the top and bottom for floor and ground avoidance. The PCBs were designed for operation of the aVIS modules on the 10-gram indoor microflyer by Zufferey et al. (2007) (section 4.1).

9.1.2 Sensor history

The adaptive vision sensor was designed to be the successor of the dynamic optical flow sensor presented in chapter 7. The unlabelled sensor area in figure 9.1 contains

the circuits that implement the facilitate-and-sample (FS) optical flow algorithm. These circuits were not usable after fabrication because of two mistakes that were made when I transferred the DOFS design from the $1.6\mu\text{m}$ to the $0.35\mu\text{m}$ fabrication process. These mistakes affected the performance of the sensor's optical flow outputs making them too noisy and unreliable for visual guidance of autonomous robots.

1. I increased the capacitive gains of the photoreceptor from $A_{pr} = 14.15$ to $A_{pr} = 23$ and the LMC circuit from $A_{lmc} = 43.57$ to $A_{lmc} = 51.2$. Due to the increased gain and in the presence of circuit noise the coupled photoreceptor and LMC showed unstable behavior where spontaneous oscillations were generated. The problem of oscillations was addressed by limiting the bias currents of the circuits' amplifiers which also limits the circuits' bandwidths.
2. I added a circuit for local competition between the FS circuits of neighboring pixels that makes the sensor's optical flow outputs unreliable. Unfortunately, I could not find a work around for addressing this problem.

Because of these problems, we decided to directly read the sensor's LMC outputs and compute optical flow off-chip on a microcontroller.

9.2 Circuits of the dynamic optical flow sensor

This section discusses the individual circuits of the adaptive vision sensor in detail.

9.2.1 Adaptive logarithmic wide-dynamic-range photoreceptor circuit

The adaptive vision sensor uses similar adaptive wide-dynamic-range photoreceptor circuit as DOFS. The only modification to the circuits is the replacement of the adaptive element proposed by Delbrück and Mead (1994, 1995) by a source-follower controlled element (M6,M7,M2) as suggested by Liu (2000) that allows to control the circuits adaptation time constant after fabrication through the bias $V_{prbfbias}$. The aVIS photoreceptor circuit is shown in figure 9.3.

The circuit amplifies temporal contrast and outputs a voltage V_{pr} .

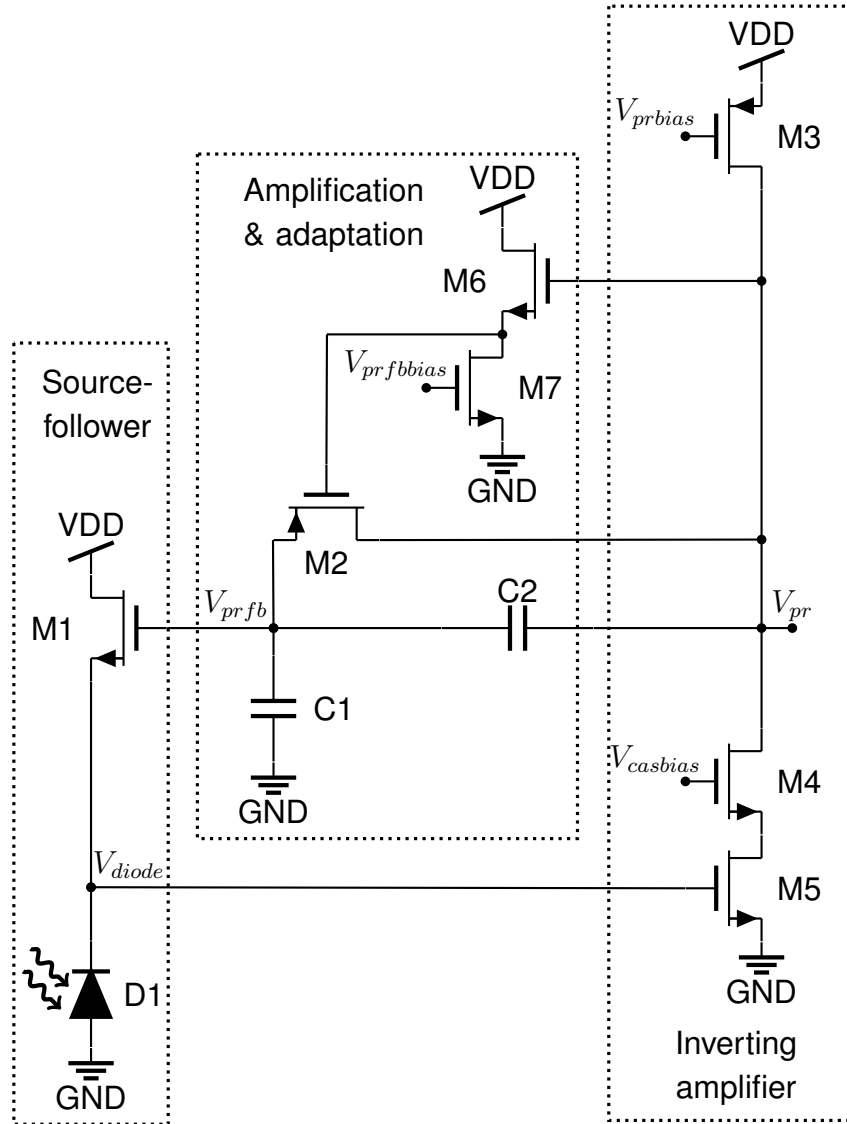


Figure 9.3: Schematic of adaptive logarithmic wide-dynamic-range photoreceptor circuit implemented as part of aVIS.

$$\frac{dV_{pr}}{dt} \approx A_{pr} \frac{U_T}{\kappa} \frac{1}{I_{ph}} \frac{dI_{ph}}{dt} \quad (9.1)$$

where κ describes the efficiency of the gate in driving the channel potential, U_T is the thermal voltage, I_{ph} is the photocurrent generated by the photodiode and A_{pr} is the gain of the capacitive amplifier (C1,C2) , that is,

$$A_{pr} = \frac{C1 + C2}{C2} \quad (9.2)$$

The gain A_{pr} for the aVIS is about 23.

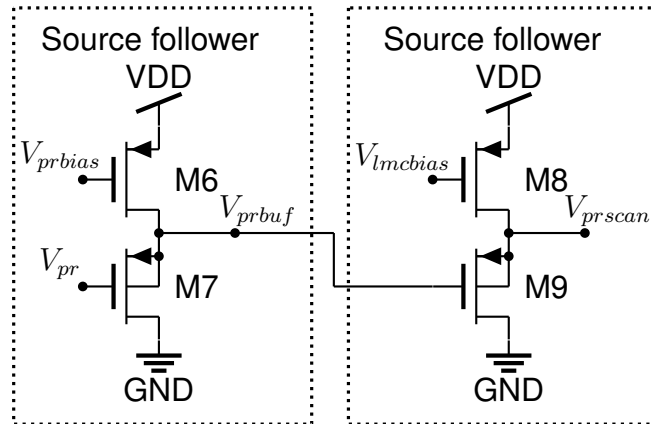


Figure 9.4: Two source follower circuits act as buffer and level shifter between the photoreceptor and LMC circuit and between photoreceptor and scanner circuit.

Similarly as in DOFS, two source follower circuits buffer the photoreceptor output signal (figure 9.4) and prevent signals from the LMC and scanner circuit from coupling back into the photoreceptor.

9.2.2 LMC circuit

The aVIS LMC circuit is the same as for DOFS. The circuit that outputs an amplified version V_{lmc} of the photoreceptor output V_{pr} is displayed in figure 9.5.

$$\frac{dV_{lmc}}{dt} = A_{lmc} \frac{dV_{pr}}{dt} \quad (9.3)$$

where the capacitive gain, $A_{lmc} = \frac{C3+C4}{C4}$ and is about 51.2 in this design.

Besides the amplification the LMC circuit implements adaptation by the two feedback elements M10-M13 and M14,M18,M19. Furthermore, the circuit removes the DC mismatch from the photoreceptor and outputs a voltage signal which DC is independent from the background light level. These properties are useful for computing the optical flow by using the Image Interpolation Algorithm on the LMC outputs.

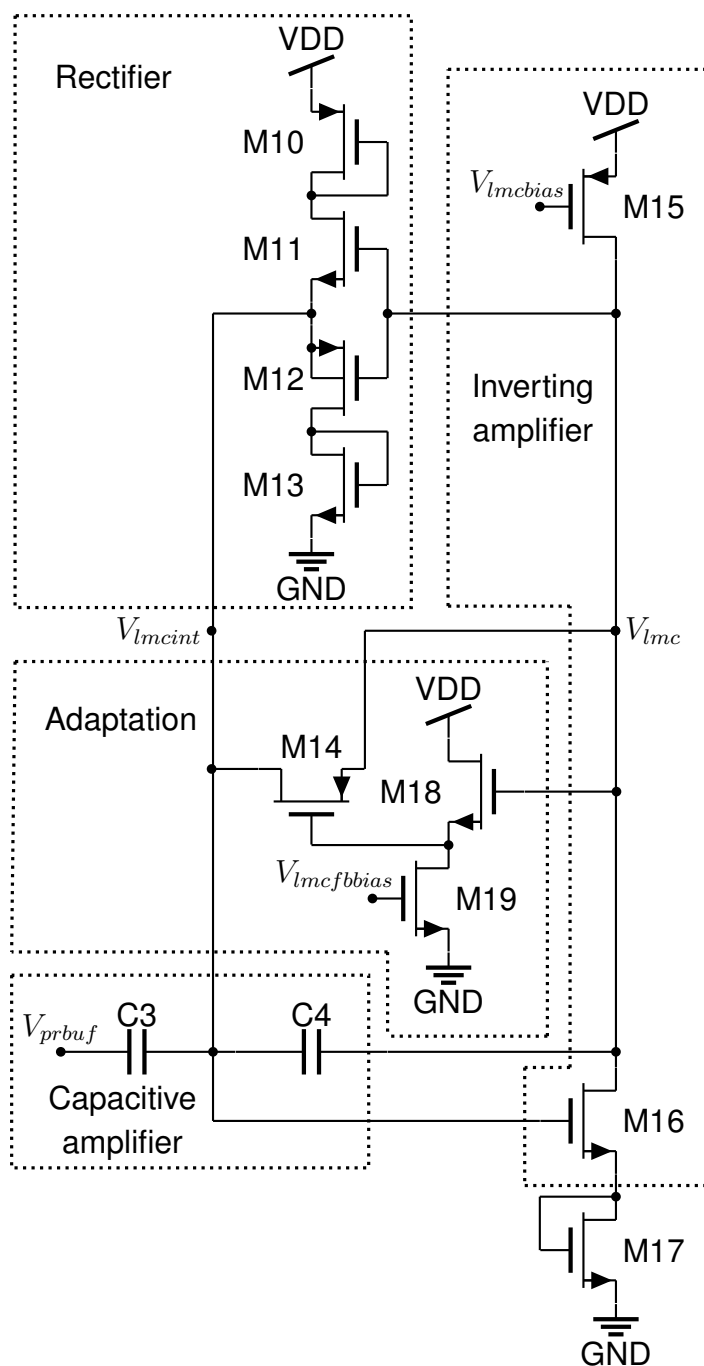


Figure 9.5: (a) Schematic of aVIS LMC circuit.

9.2.3 Scanner circuit

As in DOFS, a scanner circuit is used to read the local output values of individual pixel. I applied two modification to the original scanner circuit: (1) I modified the scanner circuit

so that several bits can be manually be loaded into the scanner's shift register and thus the scanner can read out an average over the outputs of several pixels. This option to actively load bits allows as well for the synchronization of the readout of several aVIS chips using the same digital interface, for example, in the pyramidal eye configuration. (2) Furthermore, I replaced the DOFS 5-transistor unit-gain follower circuits with wide-range follower circuits that have more gain and thus speed up the response of the scanner circuit and that can have an input and output voltage swing over the whole power supply levels.

9.3 Linear Image Interpolation Algorithm

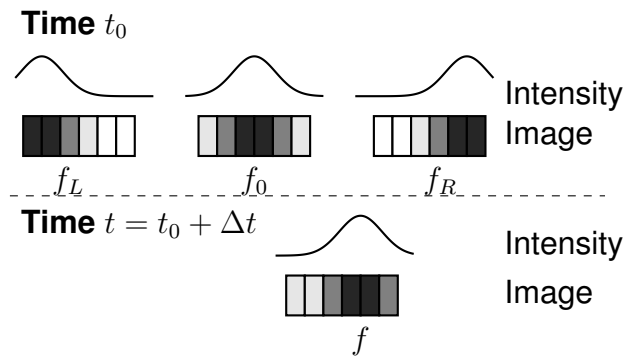


Figure 9.6: The linear I2A uses four image frames for the estimation of an optical flow value: Image frames f_0 and f are sampled successively with an inter image time $itt = \Delta t$. Image frames f_L and f_R are generated by shifting image f_0 to the left and right by one pixel, respectively.

The Image Interpolation Algorithm (I2A) by Srinivasan (1993, 1994) estimates optical flow by measuring how much the pixels' intensity values shift between two successive image frames f_0 and f as shown in Figure 9.6. On a 1-dimensional pixel array like aVIS the shift along the array $\widehat{\Delta x}$ between f_0 and f can be estimated with the help of two reference image frames with a known amount of shift. These reference images f_R and f_L can be generated by shifting the intensity values of the image frame f_0 by a known amount of pixel shift, Δx_{ref} .

$$f_R = f_0(x - \Delta x_{ref}) \quad (9.4)$$

$$f_L = f_0(x + \Delta x_{ref}) \quad (9.5)$$

If the shift between f_0 and f is sufficiently small which can be achieved by adjusting the time delay or inter-image time Δt between f_0 and f , the image shift or optical flow $\widehat{\Delta x}$ can be estimated by finding an approximation to f , called \hat{f} , that is a linear weighted combination of the reference images f_R and f_L relative to the original image f_0 . The weight corresponding to the unknown spatial shift $\widehat{\Delta x}$ is normalized to the known shift, Δx_{ref} , used to generate the reference image frames:

$$\hat{f} = f_0 + 0.5 \left(\frac{\widehat{\Delta x}}{\Delta x_{ref}} \right) (f_R - f_L) \quad (9.6)$$

The unknown $\widehat{\Delta x}$ can be found by minimizing the least square error between f and its approximation \hat{f} :

$$E = \int \Psi \cdot [f - \hat{f}]^2 dx \quad (9.7)$$

where Ψ is a function to apply a spatial low-pass filter to the image frames which Srinivasan has shown to be useful for low-noise optical flow estimation. Furthermore, Ψ can be used to select a certain patch in the image by setting all other image intensity values to zero. A typical function for Ψ is a Gaussian. By substituting \hat{f} from (9.6) into (9.7) and setting the derivative of the mean-square error E with respect to $\widehat{\Delta x}$ to zero, the unknown image shift can be obtained:

$$\frac{\widehat{\Delta x}}{\Delta x_{ref}} = 2 \cdot \frac{\int \Psi \cdot (f - f_0) (f_R - f_L) dx}{\int \Psi \cdot (f_R - f_L)^2 dx} \quad (9.8)$$

We chose the I2A for optical flow computation due to its interesting properties for embedded systems and since it was already successfully demonstrated on the indoor flyer by Zufferey et al. (2007) using a standard frame-based linear camera.

1. The I2A is a non-iterative algorithm with a closed-form solution. If Ψ is ignored, for n pixels the I2A requires only two subtractions and two multiplications per pixel as well as $(n - 1)$ additions for the spatial integration and a single division. It is thus computationally inexpensive and suitable for implementation into a microcontroller on-board a flying platform.
2. The spatial low-pass filtering of Ψ can be achieved without any computational effort by defocussing the lens with respect to the vision sensor.

3. The I2A implements spatial integration among different pixels providing proper weighting of image contrast - giving a higher weight to high contrast stimuli.

As in all gradient-based optical flow algorithms, the I2A algorithm can also become unstable in the presence of noisy images if the contrast is too low and the denominator in (9.8) is close to zero. This situation can be avoided by discarding the image frames when the denominator drops below a set threshold.

9.4 Experiments

We evaluated the adaptive vision sensors under similar conditions as the cameras originally used by Zufferey et al. (2007) for their flight experiments to test the performance of the aVIS-I2A combination. The following sections describe the experimental setup and results in more detail.

9.4.1 Experimental setup

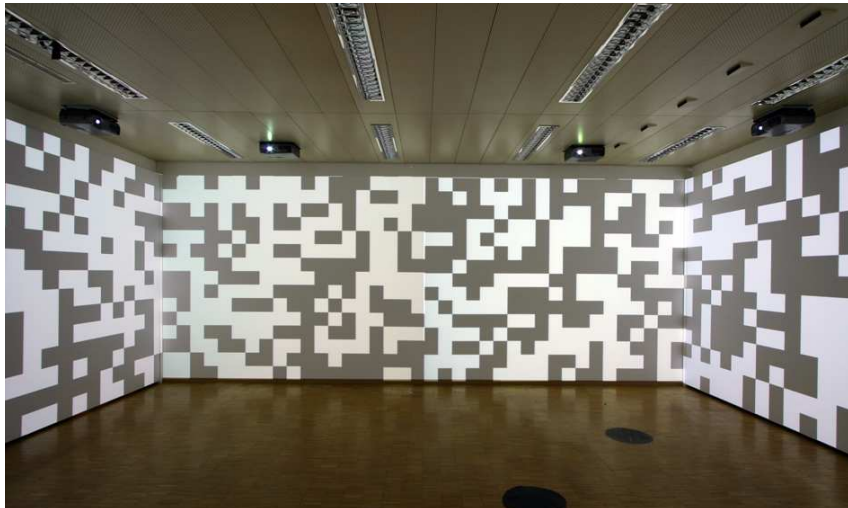


Figure 9.7: Experiments for the characterization for optical flow extraction with aVIS and the image interpolation algorithm have been carried out in the same 7m x 6m room as the flight experiments by Zufferey et al. (2007). This room is equipped with 8 projectors that can project arbitrary images on the walls. The figure showing a typical test pattern is adapted from Zufferey et al. (2007).

To evaluate the possibility of controlling a MAV with the adaptive vision sensor, we evaluated the sensors in the same room (shown in figure 9.7) that was used by Zufferey et al. (2007) for their flight experiments with their 10-gram indoor airplane. I would like to thank Jean-Christophe Zufferey for his advice on the experiments as well as Adam Klaptocz and Antoine Beyeler who together with me created the experimental setup for testing the sensors. Furthermore, I would like to thank Prof. Dario Floreano, who allowed me to do the experiments at the EPFL Laboratory of Intelligent Systems.

As discussed in section 2.3, Koenderink and van Doorn (1987) had shown that the total optical flow OF is the linear combination of a distance-independent rotational optical flow component, OF_{rot} , and a translational component, OF_{trans} , which magnitude is inversely proportional to the distance between the sensor and an object:

$$OF = OF_{trans} + OF_{rot} \quad (9.9)$$

Zufferey et al. (2007) used this fact for their flight controller that was discussed in detail in section 4.1. The controller uses the total optical flow perceived by two vision sensors and subtracts the rotational optical flow components to gain the remaining translational optical flow. The airplane's rudder is controlled by a signal OF_{Div} that is generated by subtracting the translational optical flow perceived by the left and right vision sensor. The rotational optical flow component is estimated based on the readings from a gyroscope. The controller thus fully depends on a good matching between the rotational optical flow computed from the vision sensors and the outputs of the gyroscope.

To test the compatibility of aVIS with the flight controller by Zufferey et al. (2007) the optical flow generated by the I2A based on the sensor's outputs was compared with the outputs of a rate gyroscope (IDG-500 from Invensense Inc.). During the experiments, aVIS was mounted on a rotating platform facing visual patterns that have been projected onto the walls of the experimental room (shown in figure 9.7). The LMC and photoreceptor (PR) outputs from the 60 pixels in the middle of the aVIS pixel array were recorded. For comparison we furthermore recorded images and computed optical flow from the 60 central pixels of the same off-the-shelf linear image sensor (TSL3301 from TAOS, Inc) that was used by Zufferey et al. (2007) for their flight experiments. Specifications of the TSL3301 image sensor can be found in table 9.2. For a better comparison both aVIS and TSL have been equipped with the same lens (EL20 with effective focal length $EFL=3.4\text{mm}$).

For the optical flow extraction the inter-image time, itt , for aVIS and TSL was adapted to match the theoretically expected maximal image shift between two consecutive frames

Table 9.2: Specifications of TSL3301 image sensor.

Supply voltage	3.3V
Number of pixels	102 x 1
Pixel size	85 μm x 77 μm
Pixel distance	85 μm

to a rotation rate of 300°/s. This is the maximum rotation rate measured on the target aerial platform. The aVIS was tested under the 3 conditions:

1. In the first condition called "**1 pixel shift**" the *itt* was set so that in the time between two frames the image is shifted by 1 pixel when the sensor is rotated with the maximum rotation rate of 300°/s.
2. In the second condition called "**3 pixel shift**" the *itt* was set so that in the time between two frames, the image is shifted by 3 pixels when the sensor is rotated with the maximum rotation rate of 300°/s.
3. In the last condition called "**3 pixel bins**" sensor pixels were grouped into bins of 3 pixels. The averaged PR and output values of the pixels within a bin was used as input to the I2A. So 3 pixels were treated as if they were a single one. The *itt* was set so that in the time between two frames the image is shifted by these 3 pixels when the sensor is rotated with the maximum rotation rate of 300°/s.

Our hypothesis was that averaging over several aVIS pixels would help to cancel mismatch and reduce noise in the LMC and PR output signals. Furthermore, we expected the optical flow estimates calculated from the LMC output to be of less noise and less error than the optical flow generated from the PR output because of the further amplification and the band-pass filtering of the PR signals by the LMC circuit. The biases of the circuits had been adjusted to reduce errors in the optical flow calculation due to adaptation. The lens was defocused to spatially smooth the images projected on the sensors. The sensors have been tested with the visual pattern shown in figure 9.7 using 2 different spatial frequencies.

9.4.2 Experimental results

Figures 9.8 and 9.9 show the optical flow calculated from aVIS plotted against the output of the gyroscope. Ideally all measurements should be on the diagonal line.

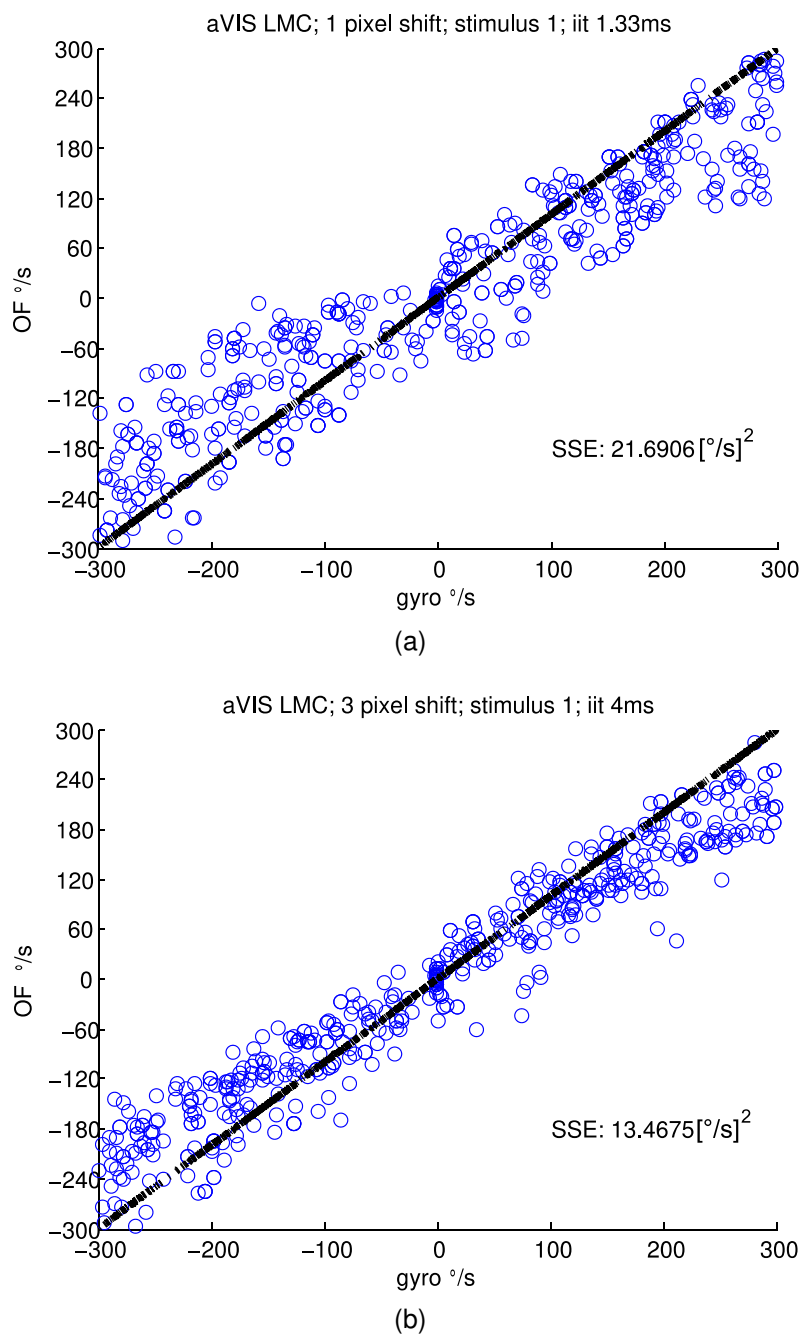


Figure 9.8: Optic flow calculated from the aVIS LMC circuit output plotted against the output of the gyroscope for the (a) 1-pixel-shift and (b) 3-pixel-shift condition in form of a scatter plot. The black dashed line shows the desired linear relationship between gyroscope and optical flow. SSE is the root mean squared error.

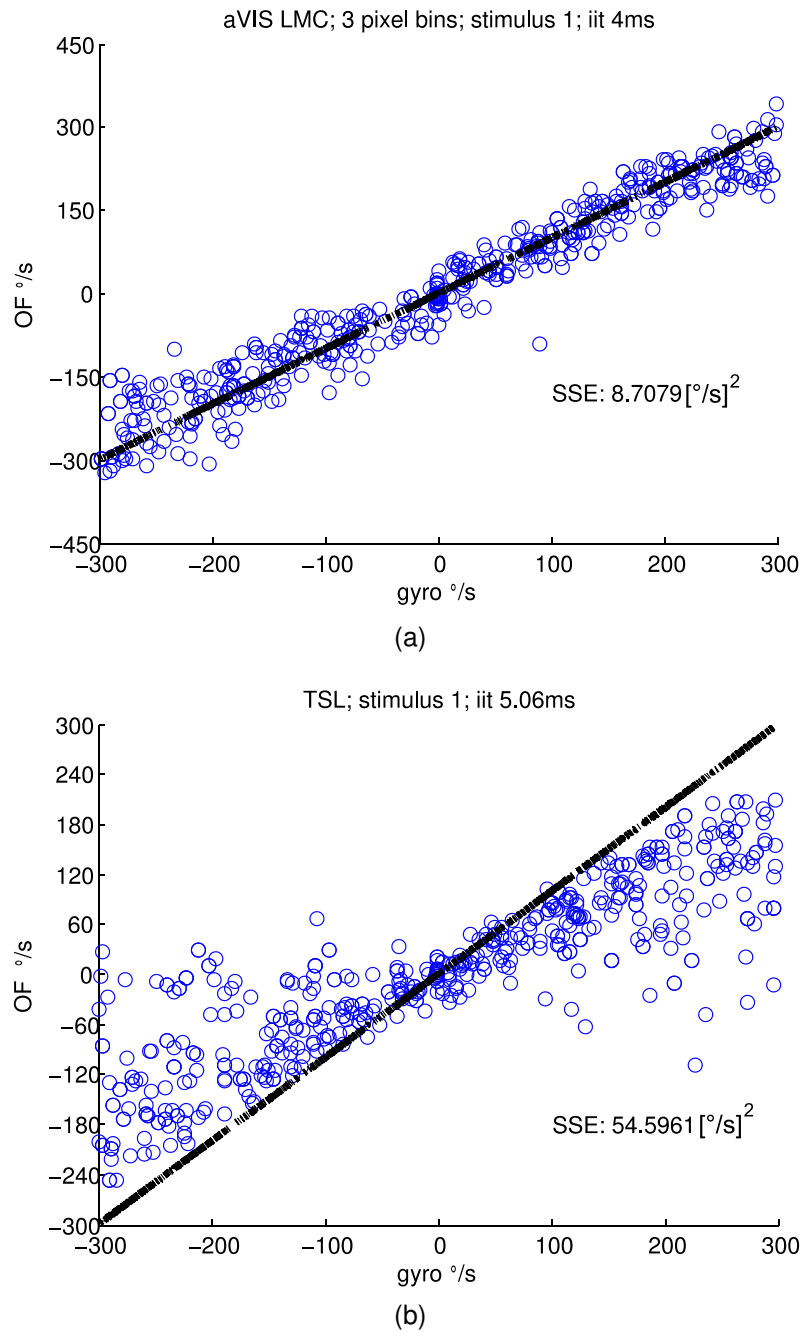


Figure 9.9: Optic flow calculated from (a) the aVIS LMC circuit output plotted against the output of the gyroscope for the 3-pixel-bin condition and (b) estimated from the TSL image sensor in form of a scatter plot. The black dashed line shows the desired linear relationship between gyroscope and optical flow. SSE is the root mean squared error.

However, due to noise, mismatch and since the precise distance between lens and sensor is not known the measurements are scattered around a line with a slope different from 1. For finding the best possible matching we did a linear least squares fit extracting the gain and offset to match optical flow and gyroscope output values. To ensure an equal contribution of optical flow values over the entire range from $-300^\circ/\text{s}$ to $300^\circ/\text{s}$ we separated the measurements into 10 bins and selected 30 OF-gyroscope value pairs for calculating the least squares fit.

Figures 9.10 to 9.15 show samples from the gyroscope and optical flow time traces when aVIS was rotated in front of the visual stimulus patterns 1 and 2. The plots reveal the differences in the matching for the different test conditions. A good matching between gyroscope and optical flow output seems to be achieved for instance by the experiment which results are shown in figure 9.15a.

To quantify the matching between gyroscope and I2A optical flow data we again binned data samples into 10 equally sized bins of optical flow velocity ranges between $-300^\circ/\text{s}$ and $300^\circ/\text{s}$. For each bin we randomly sampled 30 values from the time traces and calculated the root mean squared error (RMSE) between the mean optical flow value per bin and the desired linear optical flow to gyroscope relationship. The results are shown in figures 9.16 to 9.21.

For comparison we also recorded the optical flow generated by the TSL camera. The results are shown in figure 9.22.

9.5 Discussion

The results of the different experiments are summarized in figure 9.23. From the RMSE measurements we conclude that when using aVIS in combination with the I2A the best results are obtained by binning pixels and averaging their LMC output signals.

The experiments show that the smaller spatial frequencies in stimulus pattern 2 were better suited for the 3-pixel-binning and 3-pixel-shift than for 1-pixel-shift condition. This is probably because the I2A prefers many smooth gradients across the sensor array. When limiting the image shift to a maximum of a single sensor pixel, the pixel's FOV is too small to receive a sufficient gradient and resolve different image velocities with good resolution and low noise. Furthermore, the mismatch has larger effect for the 1-pixel-shift condition since no averaging between pixels takes place before the optical flow calculation. The spatial optical flow integration of the I2A is of benefit but a combination of the two integration techniques is of more advantage.

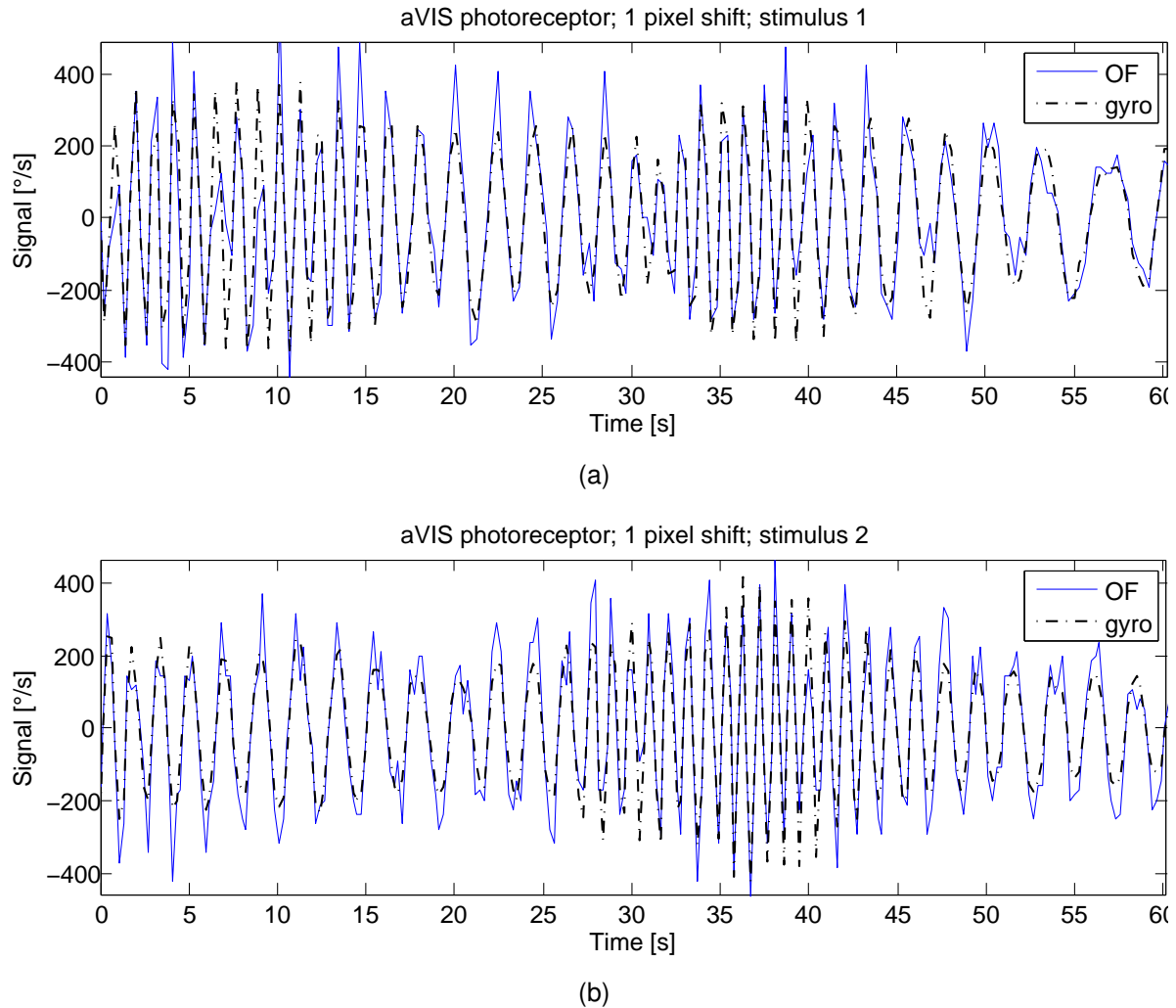


Figure 9.10: Timing diagram of optical flow and gyroscope output signals. The optical flow output was corrected by the gain and offset found with a least squares fit to match the gyroscope as good as possible. Time traces are shown for (a) the visual stimulus pattern 1 and (b) stimulus 2. Optic flow was calculated from the aVIS photoreceptor outputs. The optical flow follows the gyroscope best when the rotation rate changes smoothly but tends to overshoot at high accelerations and decelerations.

The results also clearly indicate that the LMC output is better suited for optical flow computation than the outputs of the TSL sensor and as expected than the aVIS photoreceptor outputs. The only exception where the LMC performs worst is for stimulus 2 and the 1-pixel shift condition. The advantage from the optical flow generated from the LMC output over those estimated from the TSL sensor could be partially caused by the better spatial resolution of aVIS. The LMC outputs are better suited for optical flow calculation than the photoreceptor outputs because the LMC high-pass filters and

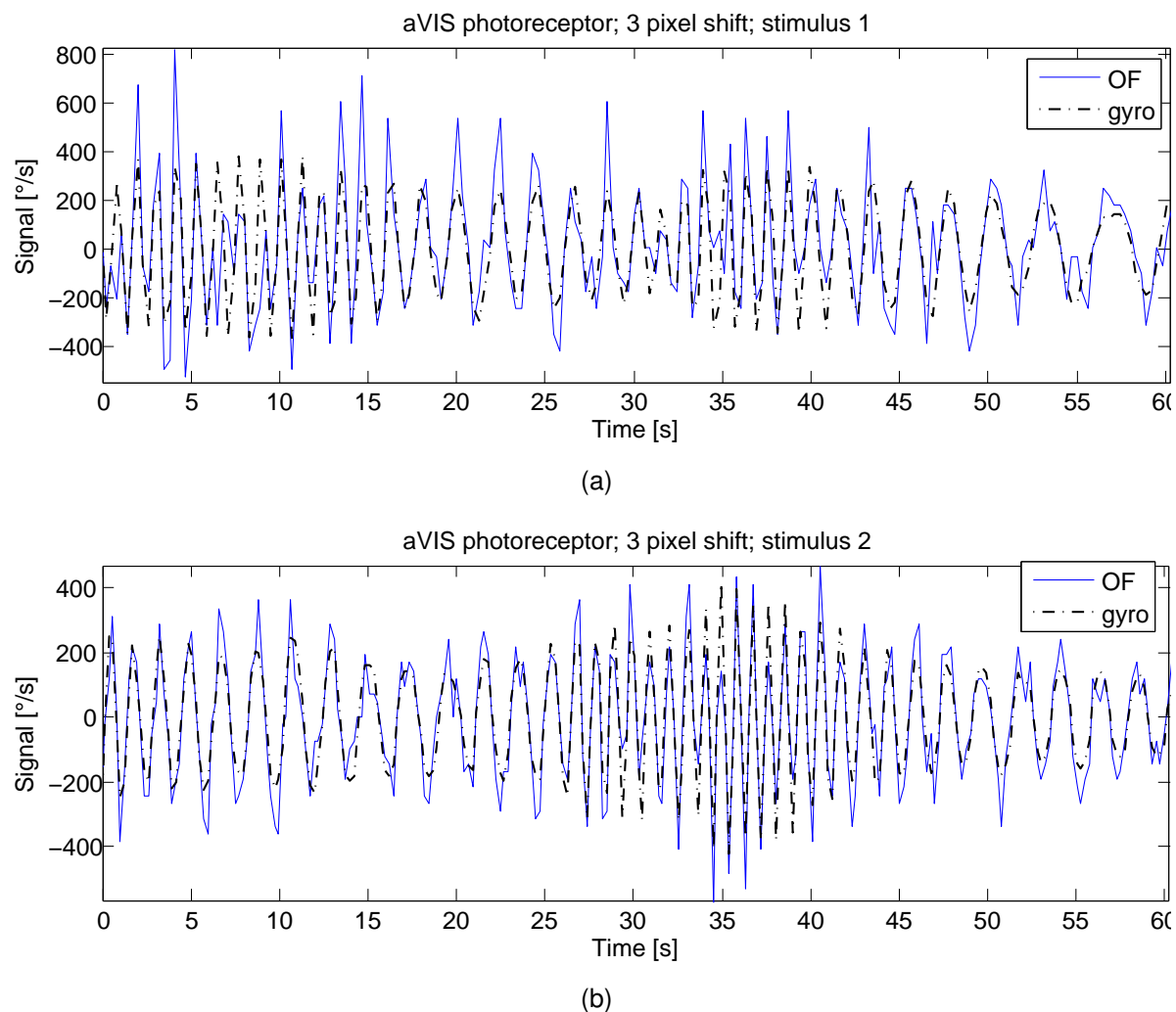


Figure 9.11: Timing diagram of optical flow and gyroscope output signals. The optical flow output was corrected by the gain and offset found with a least squares fit to match the gyroscope as good as possible. Time traces are shown for (a) the visual stimulus pattern 1 and (b) stimulus 2. Optic flow was calculated from the aVIS photoreceptor outputs.

amplifies the PR output. Thus the LMC circuit is removing the DC mismatch of the photoreceptor outputs but also makes the signals better suited for the conversion by the 10-bit analog-to-digital converter that has an input range from 0 to 3.3V. The LMC bias controlling the current in the inverting amplifier $V_{lmc\,bias}$ was set so that high frequencies were filtered - a further advantage over the photoreceptors' outputs.

Further improvements of the optical flow quality is possible by adjusting the range of analog-to-digital converter (ADC) to fit to maximum peak-to-peak amplitude of the LMC

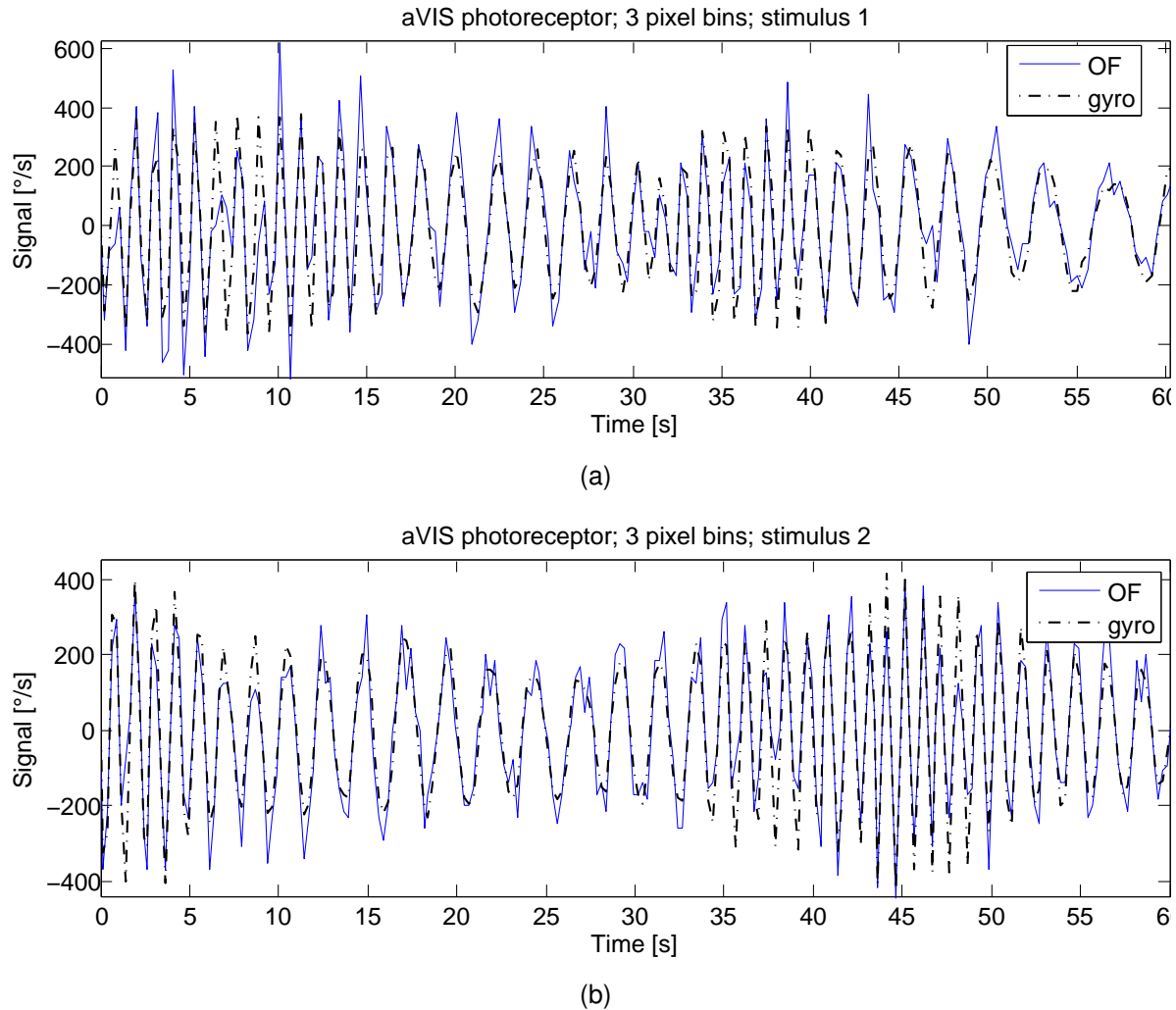


Figure 9.12: Timing diagram of optical flow and gyroscope output signals. The optical flow output was corrected by the gain and offset found with a least squares fit to match the gyroscope as good as possible. Time traces are shown for (a) the visual stimulus pattern 1 and (b) stimulus 2. Optic flow was calculated from the aVIS photoreceptor outputs.

output signal so that resolution of the ADC is optimally used. Furthermore, the system would benefit from the addition of a sample and hold circuit to the LMC output. Currently the values of the pixels can change while pixels are scanned due to image motion. So for large sensor arrays when the time deviation between sampling pixels cannot be neglected anymore the moving stimuli or sensor can generate considerable distortion in the optical flow measurements. With a simple sample-and-hold circuit this can be avoided.

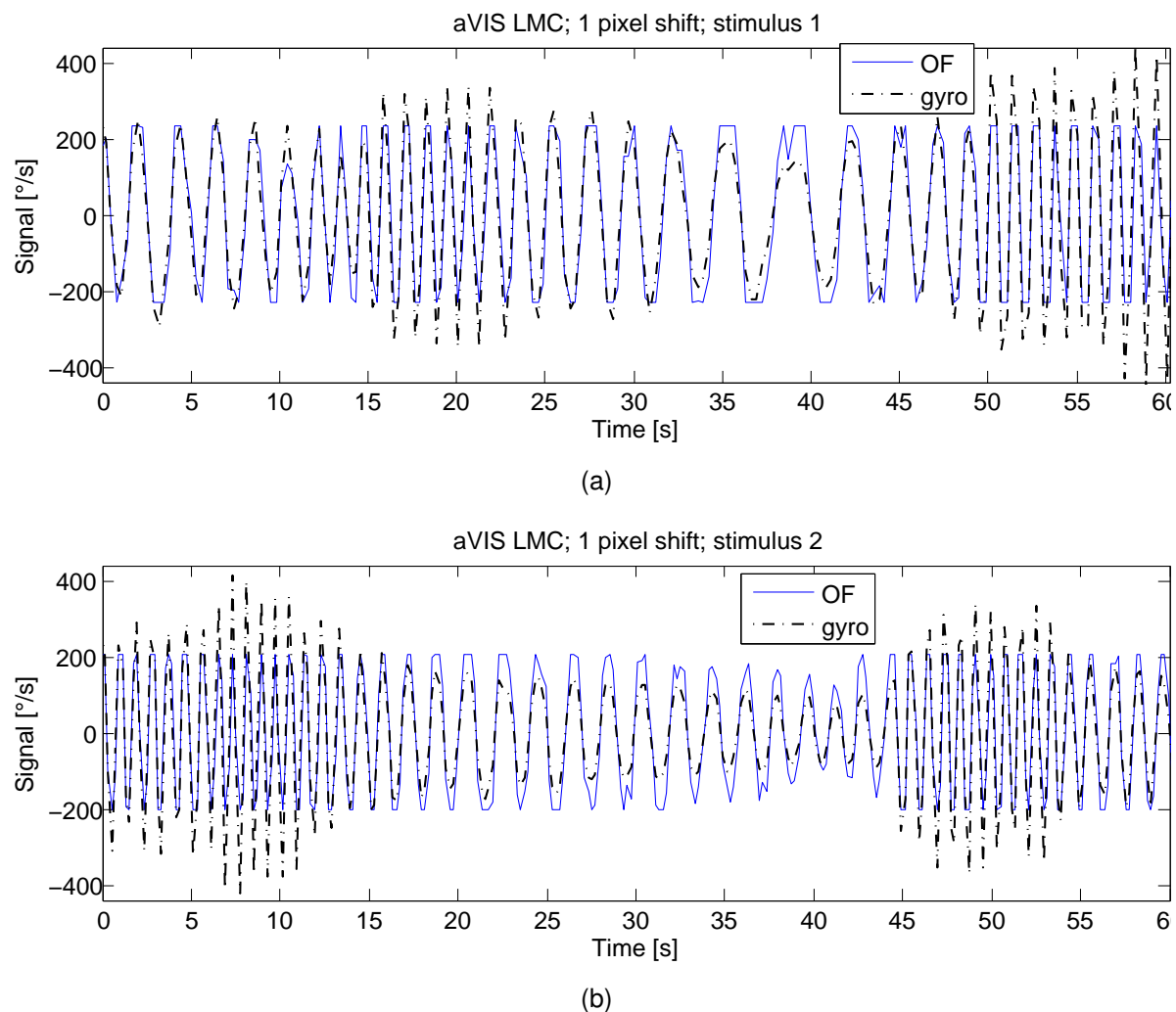


Figure 9.13: Timing diagram of optical flow and gyroscope output signals. The optical flow output was corrected by the gain and offset found with a least squares fit to match the gyroscope as good as possible. Time traces are shown for (a) the visual stimulus pattern 1 and (b) stimulus 2. Optic flow was calculated from the aVIS LMC outputs.

Overall our experimental results show that a combination of I2A and the continuous-time adaptive image sensor is possible. The RMSE measurements make confident that autonomous guidance of a MAV like the indoor flyer by Zufferey et al. (2007) are within reach using our adaptive vision sensor.

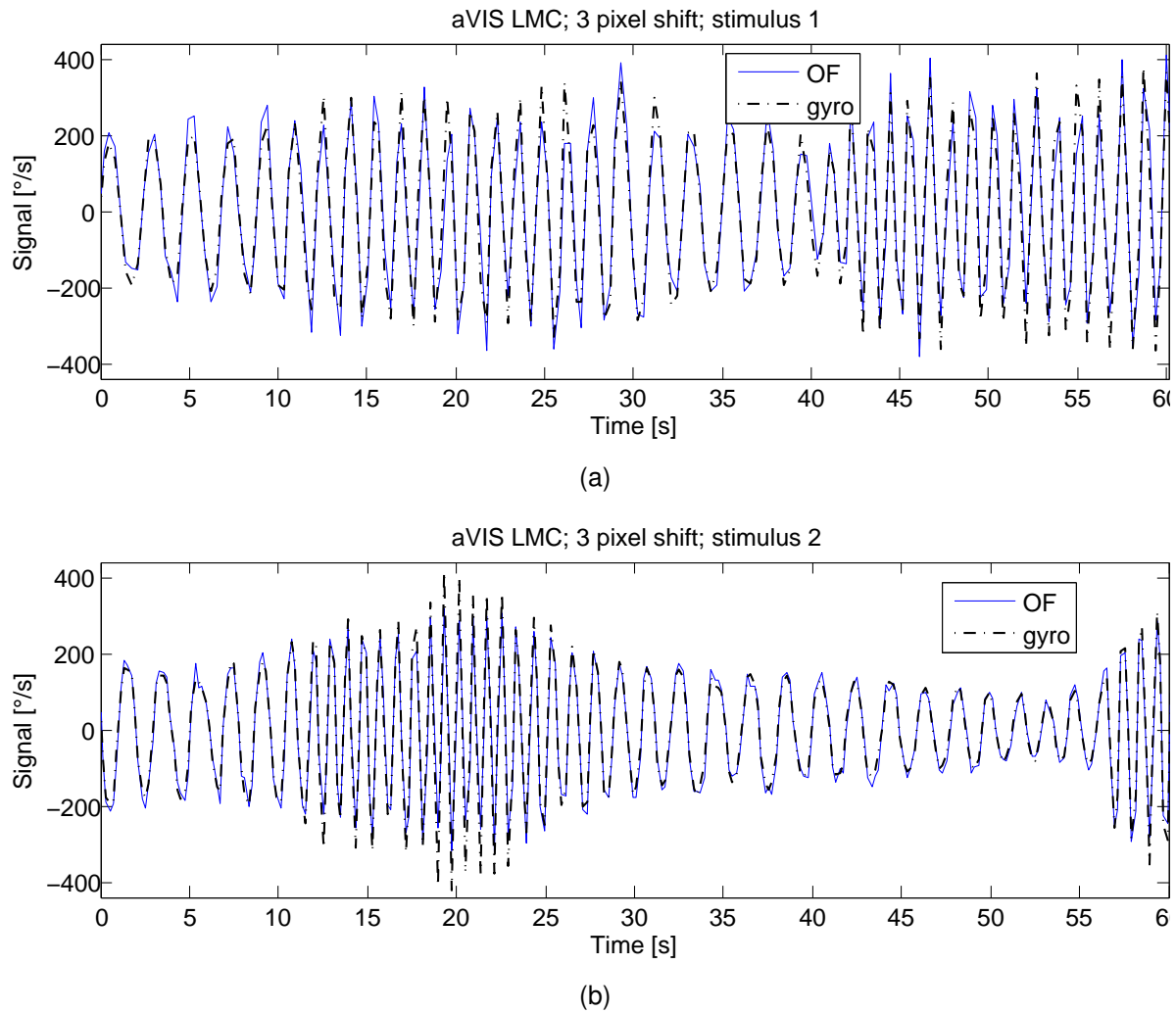


Figure 9.14: Timing diagram of optical flow and gyroscope output signals. The optical flow output was corrected by the gain and offset found with a least squares fit to match the gyroscope as good as possible. Time traces are shown for (a) the visual stimulus pattern 1 and (b) stimulus 2. Optic flow was calculated from the aVIS LMC outputs.

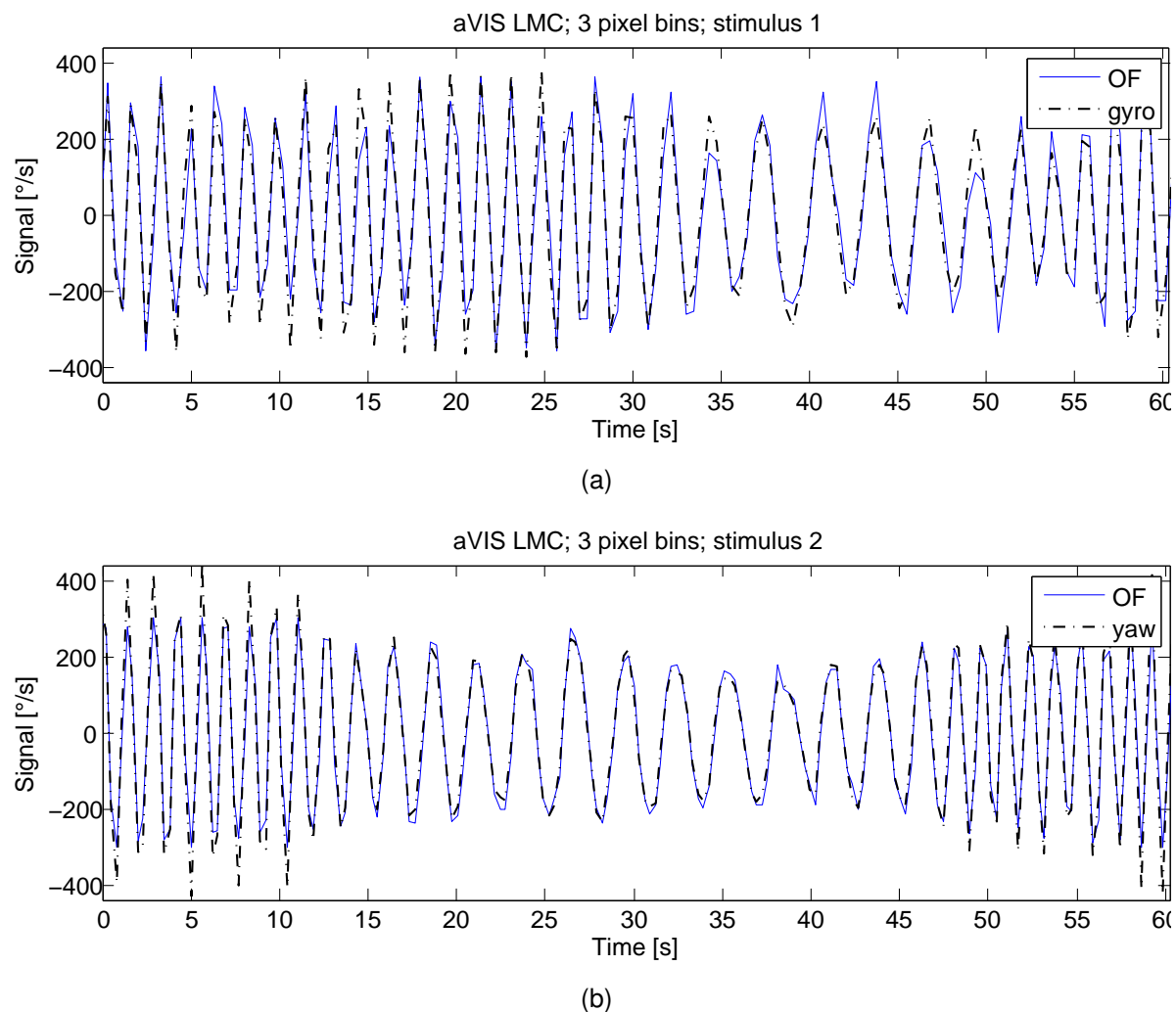


Figure 9.15: Timing diagram of optical flow and gyroscope output signals. The optical flow output was corrected by the gain and offset found with a least squares fit to match the gyroscope as good as possible. Time traces are shown for (a) the visual stimulus pattern 1 and (b) stimulus 2. Optic flow was calculated from the aVIS LMC outputs. When averaging the LMC output from 3 pixels the calculated optical flow closely matches the output from the gyroscope.

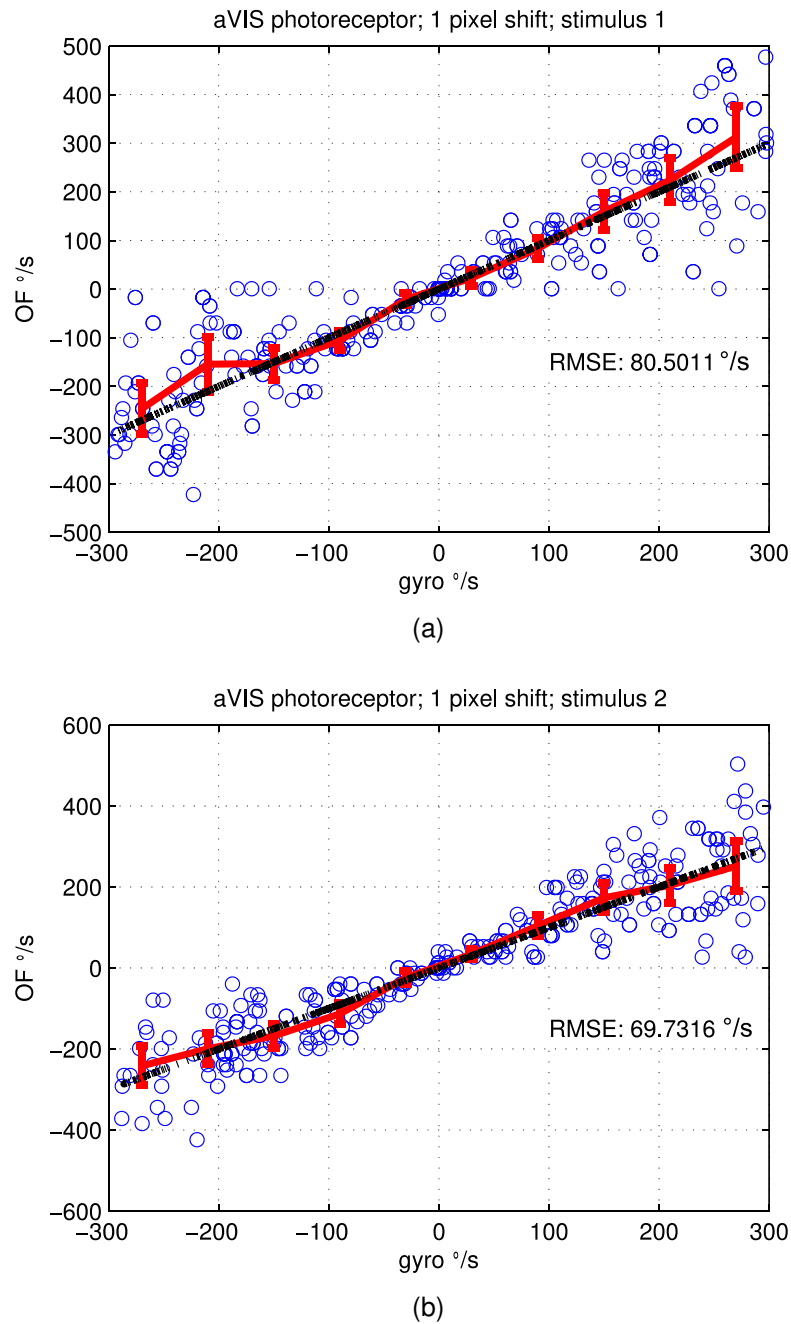


Figure 9.16: Optic flow estimates plotted against output of gyroscope for a 1 pixel shift. The optical flow output extracted from the aVIS photoreceptor was corrected by the gain and offset found with a least squares fit to match the gyroscope output. For all plots from figures 9.16 to 9.21, the range of rotation rates from $-300^{\circ}/s$ and $300^{\circ}/s$ was separated into 10 bins. The red error bar plots shows the mean and standard deviation of the 30 value pairs per bin randomly selected from the time traces. The black dashed line indicates the desired linear relationship between optical flow and gyroscope outputs.

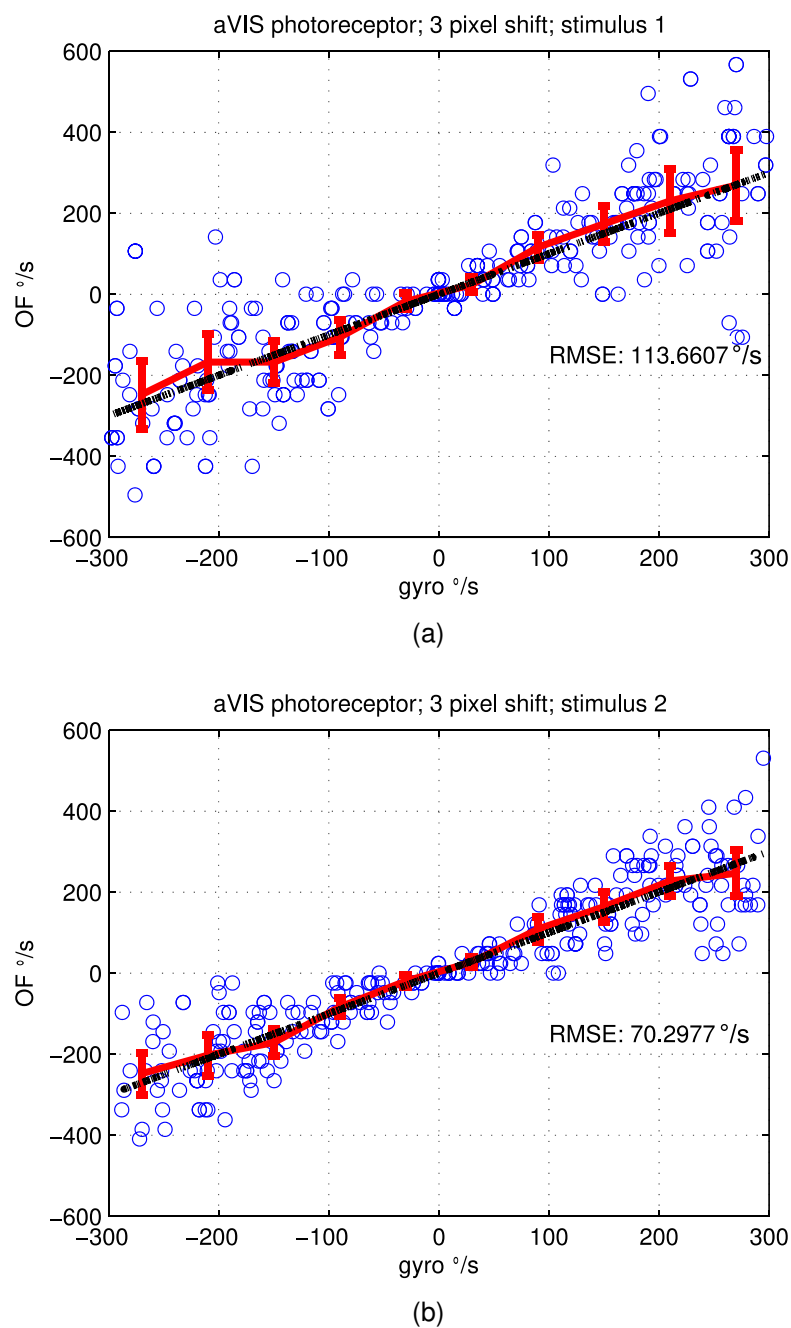


Figure 9.17: Optic flow estimates plotted against output of gyroscope for a 3 pixel shift. The optical flow output extracted from the aVIS photoreceptor was corrected by the gain and offset found with a least squares fit to match the gyroscope output. The red error bar plots shows the mean and standard deviation of the 30 value pairs per bin randomly selected from the time traces. See caption of figures 9.16 for more details.

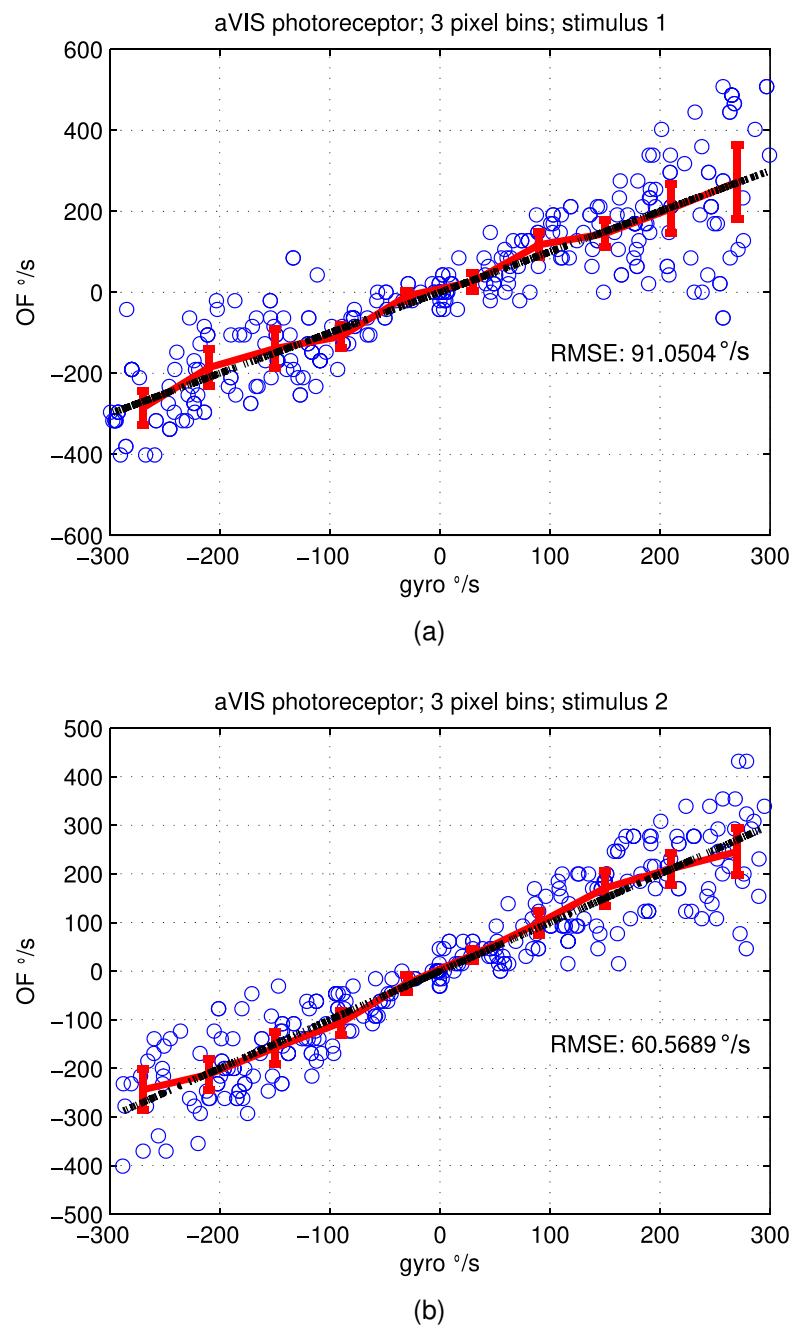


Figure 9.18: Optic flow estimates plotted against output of gyroscope for a 3 pixel bin average. The optical flow output extracted from the aVIS photoreceptor was corrected by the gain and offset found with a least squares fit to match the gyroscope output. The red error bar plots shows the mean and standard deviation of the 30 value pairs per bin randomly selected from the time traces. See caption of figures 9.16 for more details.

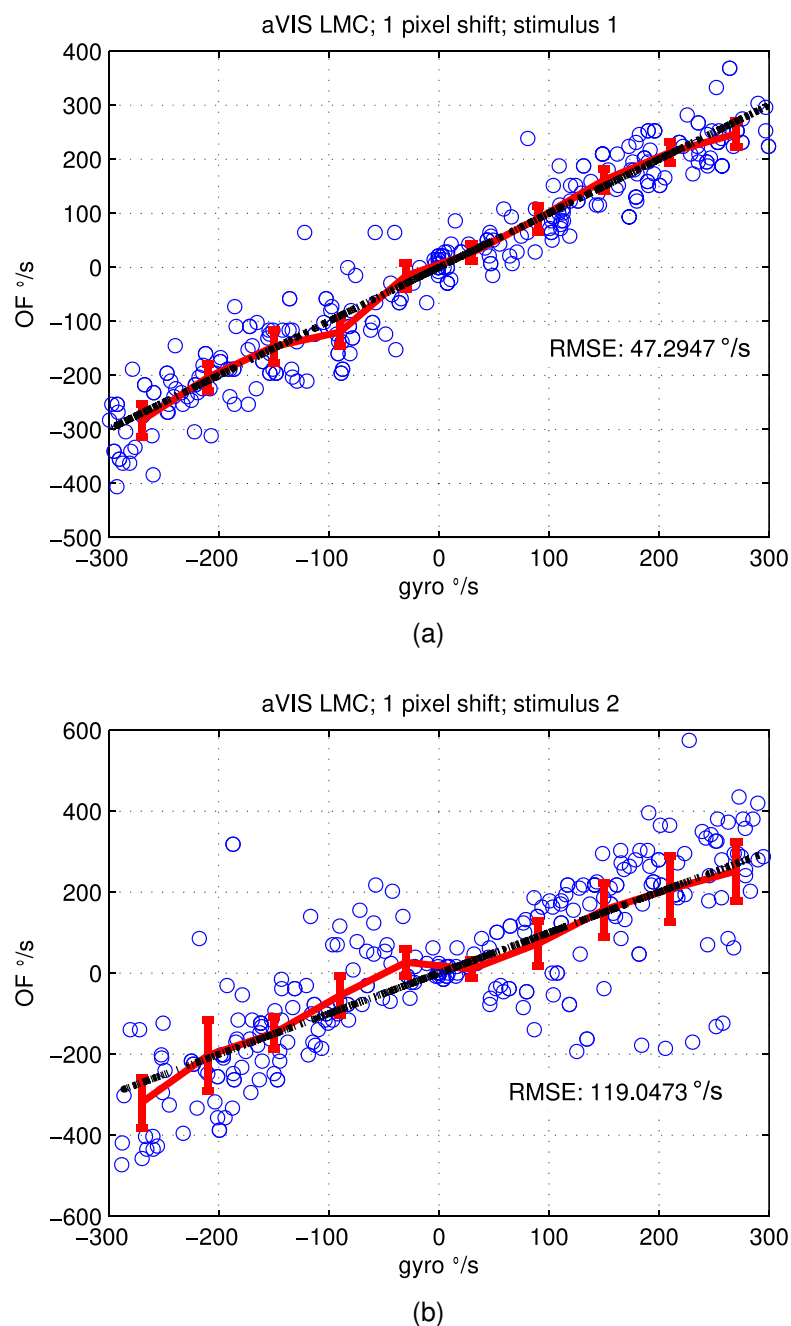
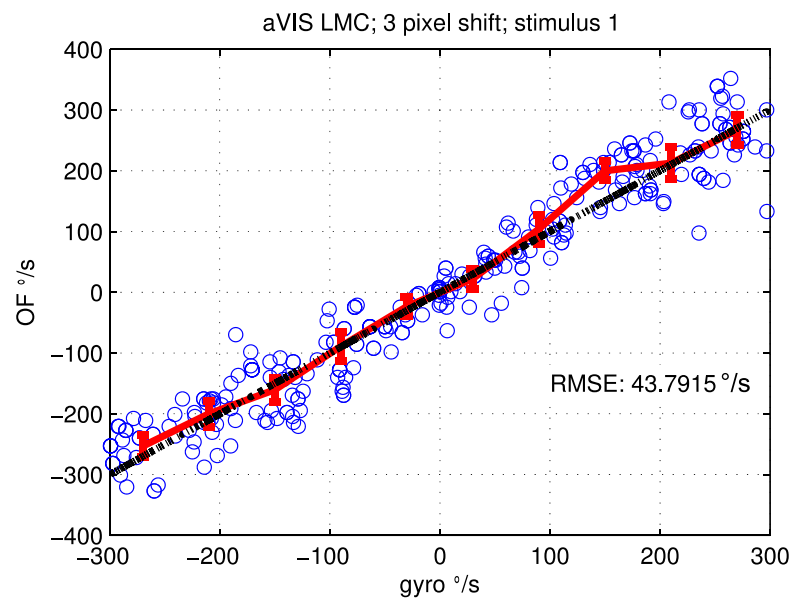
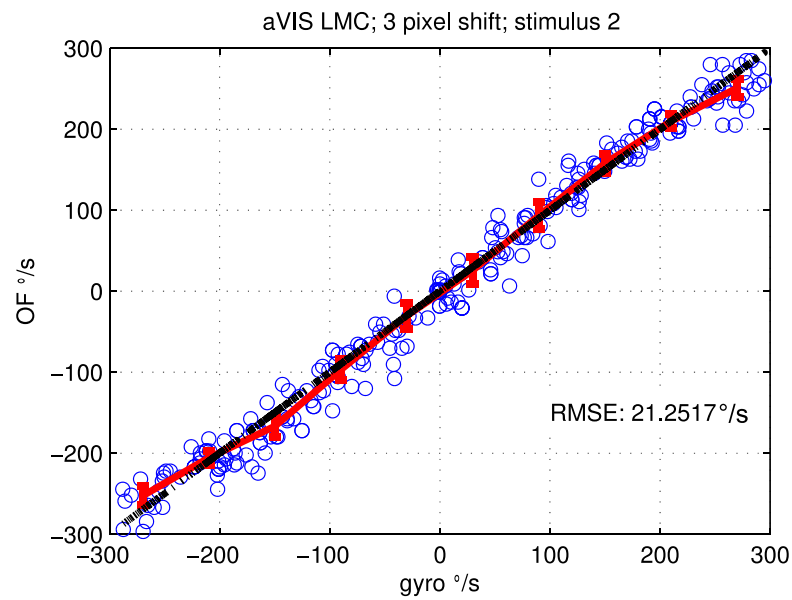


Figure 9.19: Optic flow estimates plotted against output of gyroscope for a 1 pixel shift. The optical flow output extracted from the aVIS LMC circuit was corrected by the gain and offset found with a least squares fit to match the gyroscope output. The red error bar plots shows the mean and standard deviation of the 30 value pairs per bin randomly selected from the time traces. See caption of figures 9.16 for more details.

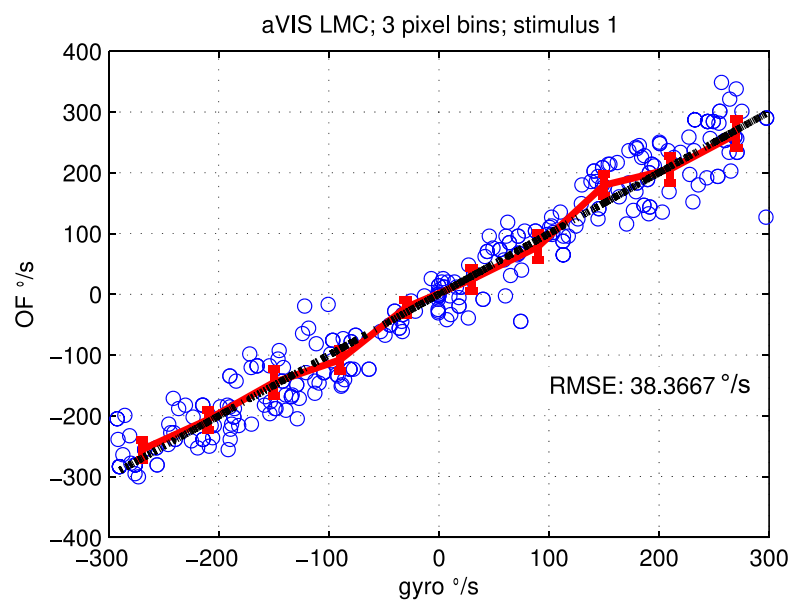


(a)

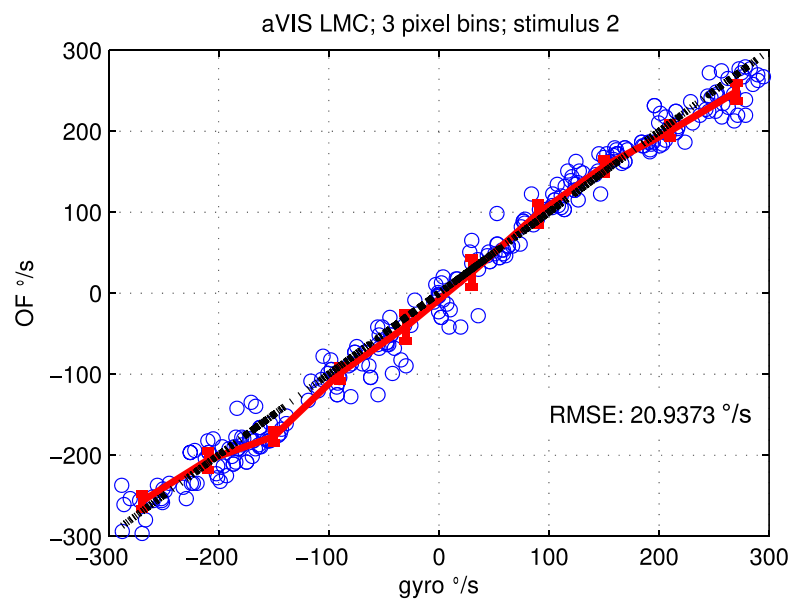


(b)

Figure 9.20: Optic flow estimates plotted against output of gyroscope for a 3 pixel shift. The optical flow output extracted from the aVIS LMC circuit was corrected by the gain and offset found with a least squares fit to match the gyroscope output. The red error bar plots shows the mean and standard deviation of the 30 value pairs per bin randomly selected from the time traces. See caption of figures 9.16 for more details.

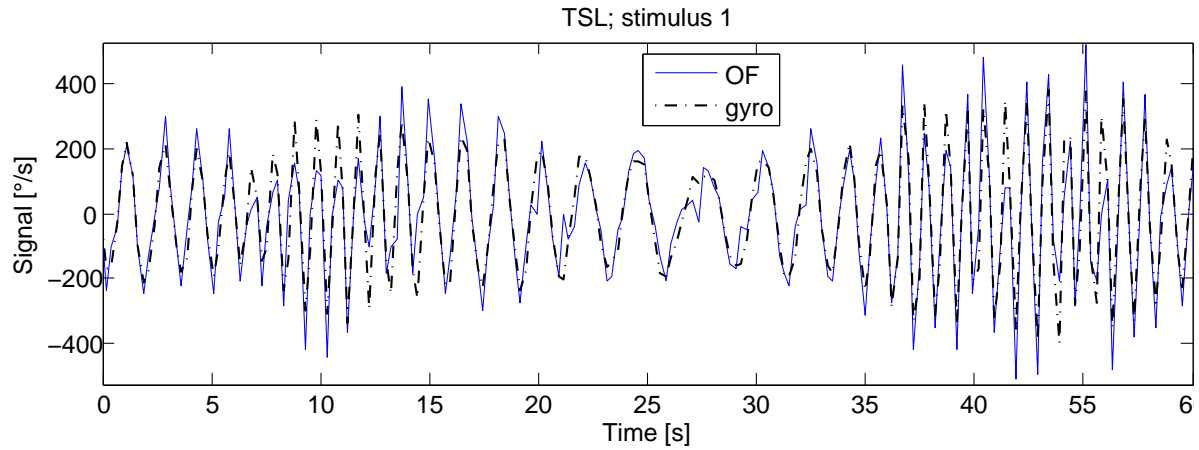


(a)

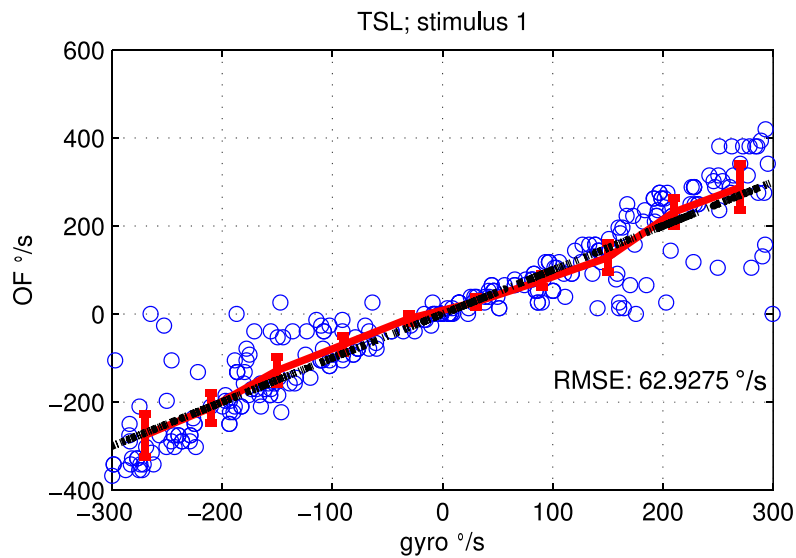


(b)

Figure 9.21: Optic flow estimates plotted against output of gyroscope for a 3 pixel bin average. The optical flow output extracted from the aVIS LMC circuit was corrected by the gain and offset found with a least squares fit to match the gyroscope output. The red error bar plots shows the mean and standard deviation of the 30 value pairs per bin randomly selected from the time traces. See caption of figures 9.16 for more details.



(a)



(b)

Figure 9.22: Experimental results from the off-the-shelf TSL image sensor. (a) Timing diagram of optical flow calculated from TSL sensor and gyroscope output. (b) Optic flow estimates plotted against output of gyroscope. The red error bar plots shows the mean and standard deviation of the 30 value pairs per bin randomly selected from the time traces. See caption of figures 9.16 for more details. The optical flow values in (a) and (b) were adjusted with the gain and offset calculated with a least squares fit.

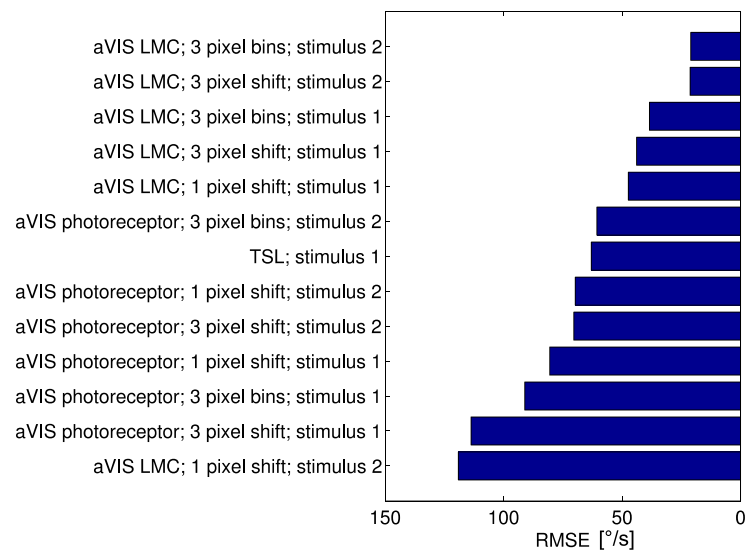


Figure 9.23: Overview of root mean squares error measurements from the two sensors and different experimental conditions.

Chapter 10

MAVs and controller for efficient use of optical flow sensors

Abstract

As presented in chapter 4 state-of-the-art optical flow controlled MAVs combine different arrangements of optical flow sensors and rate gyroscopes for (semi-) autonomous flight. The outputs of rate gyroscopes are used to estimate and remove the distance-independent optical flow component caused by self-rotation of a MAV from the total optical flow that is measured with optical flow sensors. The remaining translational optical flow is known to have a magnitude that is inversely proportional to the relative distance between the MAV and the objects in the MAV's environment and can thus be used for obstacle avoidance and altitude control.

The present chapter demonstrates that this state-of-the-art control strategy does not optimally use optical flow sensors. The presented analysis suggests that optical flow sensors can be used more efficiently if MAV and controller mimic the flight behavior found in insects more closely. While most of the optical flow controlled MAVs presented in the literature support slow turning maneuvers insects like flies and bees exploit rapid turns - so-called flight saccades. The study presented in this chapter shows that the ability to perform flight saccades does not only improve the maneuverability of MAVs but also allow the controller to exploit optical flow sensors more efficiently.

10.1 Theoretical analysis on wall avoidance with optical flow guided MAVs

It is a well-known fact that one should not try to gain a control signal of small amplitude by subtracting two large signals from noisy sources. When subtracting two signals of large amplitude the noise in the signals is not reduced and thus the signal-to-noise ratio of the difference is low. So when calculating the distance-dependent translational optical flow component OF_{trans} by subtracting the noisy output of a gyroscope OF_{rot} from the noisy total optical flow perceived by an optical flow sensor OF_{total} , the signal-to-noise ratio of the translational optical flow component is higher the lower the amplitude of the gyroscope output and the more equal OF_{total} and OF_{trans} .

This section analyzes the typical ratio between the optical flow components OF_{trans} and OF_{rot} that a MAV is confronted with during wall avoidance. To answer this question we need to find an estimate of the highest bound of the ratio $OF_{\text{trans,max}}/OF_{\text{rot,max}}$. So we have to find an estimate for $OF_{\text{trans,max}}$ which represents the maximum translational optical flow that a sensor still needs to measure reliably and to relate this to the maximum optical flow that will be caused by the self-rotation of the sensor, $OF_{\text{rot,max}}$.

Scenario: wall avoidance

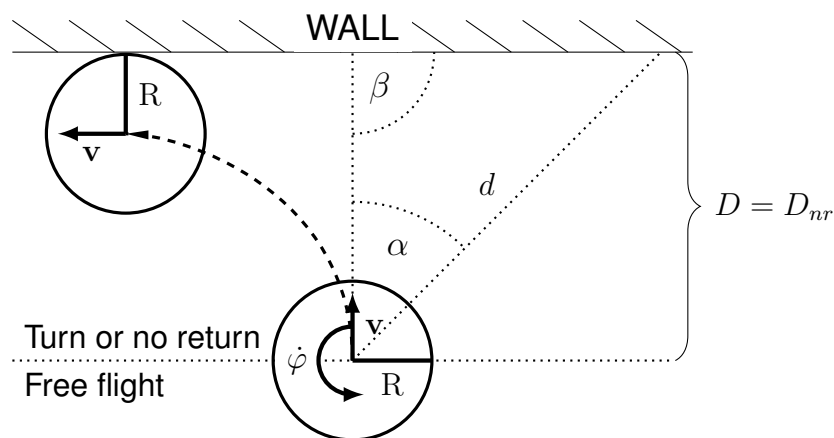


Figure 10.1: An optical flow guided MAV is approaching a WALL with a forward velocity v . At a minimum distance $D = D_{nr}$ the MAV initiates a turning maneuver with a rotation rate $\dot{\phi}$. α is the angle between the MAV's direction of translation and the direction in which the MAV's optical flow sensor is pointing towards the wall. R is the MAV's collision radius.

Obstacle or wall avoidance of an optical flow controlled MAV can be studied with the scenario shown in figure 10.1 which depicts a MAV displayed as a circle approaching a wall at an angle $\beta=90^\circ$ and then turning away from the wall. The distance between the center of the MAV and the wall is called D . In figure 10.1 the MAV is shown at two positions: (1) at the distance-before-no-return, $D = D_{nr}$, - the minimum wall distance at which the MAV has to initiate a turning maneuver to still be able to avoid a collision with the wall - and (2) after a successful turn moving parallel to the wall at the minimum distance $D = R$ that causes no collision. R is the collision radius around the MAV.

For this analysis a flying platform and flight controller similarly to the quadcopter presented by Zingg et al. (2010) and the microflyer platform by Zufferey et al. (2007) (chapter 4) is assumed. So the MAV's self-motion can be described by a forward velocity v that for example can be a linear superposition of thrust and lift and an angular rate $\dot{\varphi}$ that represents the MAV's rotation rate. Note that figure 10.1 does not need to depict a top view of the wall-avoidance scenario but can be a view from any angle of the 3-dimensional flight trajectory of the MAV shown as a dashed line. So the example could depict an airplane that both rolls and yaws at a constant rate after passing $D = D_{nr}$.

The MAV is assumed to be equipped with one or several optical flow sensors that share the same ideal properties as the spherical optical flow sensor described in section 2.3 except for the fact that the sensor in the present scenario has only a very small field of view and outputs a single optical flow value $OF_{total} = OF_{trans} + OF_{rot}$. The optical flow sensor faces the wall in a direction that differs from the MAV's direction of translation by the angle α . The distance between the sensor and the wall in the direction of the sensor is d .

In the given 2-dimensional scenario the translational optical flow OF_{trans} perceived by the sensor that was formally described by (2.10) can be simplified to

$$OF_{trans} = \frac{v}{d} \cdot \sin(\alpha) \quad (10.1)$$

and the rotational optical OF_{rot} flow formerly described by (2.11) can be simplified to

$$OF_{rot} = \dot{\varphi} \quad (10.2)$$

The total optical flow OF_{total} induced by egomotion of the MAV which as presented in section 2.3 is a linear combination of the two optical flow components caused by rotation and translation is given by

$$OF_{\text{total}} = OF_{\text{trans}} + OF_{\text{rot}} \quad (10.3)$$

$$OF_{\text{total}} = \frac{v}{d} \cdot \sin(\alpha) + \dot{\varphi} \quad (10.4)$$

From (10.4) one can estimate the wall distance in the direction of the optical flow sensor by

$$d = \frac{v}{OF_{\text{total}} - OF_{\text{rot}}} \cdot \sin(\alpha) \quad (10.5)$$

when the MAV's velocity v is known.

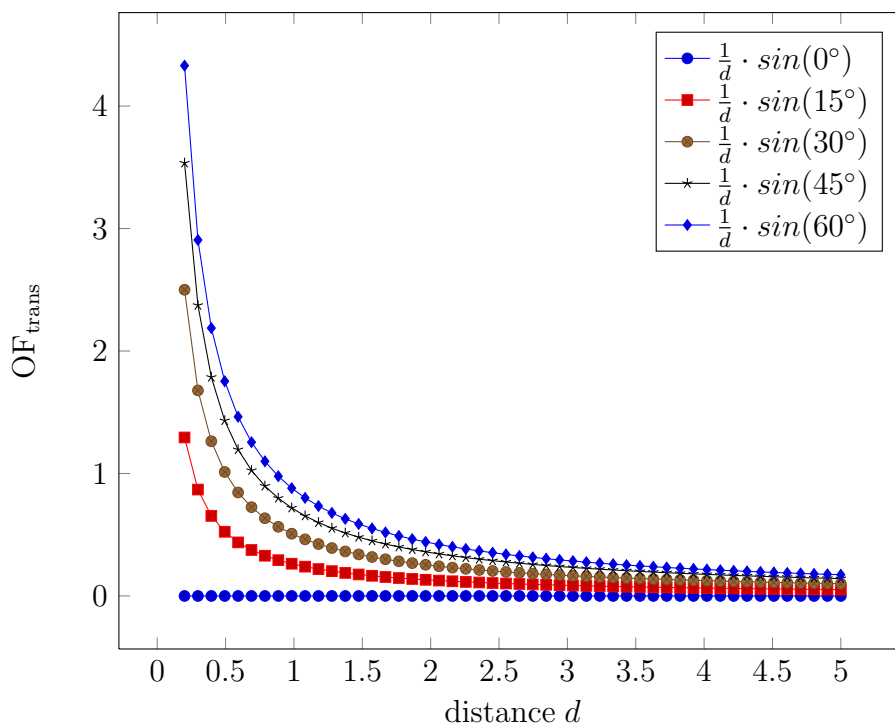


Figure 10.2: Translational optical flow, OF_{trans} , measured by a perfect spherical optical flow sensor when approaching a wall at a constant velocity v . The different curves correspond to different distances d and different angles α between the sensor's direction of translation and the direction in which the sensor is pointing. d is given in multiples of the velocity v .

Figure 10.2 displays the amount of translational optical flow perceived by a sensor when the MAV is approaching the wall at a constant velocity. Optical flow values are calculated at different distances d and for different angles α . Figure 10.2 shows that the optical flow sensor will measure more optical flow, the bigger the angle between the

MAV's direction of translation and the direction in which the sensor is pointing. However, one has to keep in mind that for large angle α , objects in front of the MAV that can lead to a collision cannot be detected.

Relation of maximum rotational and translational optical flow perceived on a MAV

From (10.2) it is known that for the considered spherical sensor the maximal rotational optical flow is equal to the maximal rotation rate of the observer. Only the optical flow generated during translation is dependent on the distance d between sensor and wall (equation (10.1)) and reaches its maximum when the distance is at its minimum. However, due to the limited rotation rate $\dot{\phi}$ of the MAV there is a certain distance $D = D_{nr}$ at which a turning maneuver needs to be initiated (figure 10.1). If at this distance-before-no-return D_{nr} the MAV does not start turning immediately it will collide with the wall. So D_{nr} is the minimum distance that at least needs to be detected and also the point where the maximal translational optical flow that still needs to be measured reliably by the sensor is measured.

For an observer traveling in an airplane with a constant thrust velocity v and a collision radius R (figure 10.1) the minimum distance at which the turning maneuver needs to be initiated, D_{nr} , is given by

$$D_{nr} = v \cdot t_{rot} + R \quad (10.6)$$

where R is the minimum distance that the center of the airplane has to keep to the wall to avoid a collision and t_{rot} is the minimum time to rotate the MAV by the angle β .

Depending on from which angle β the MAV is approaching the wall, the distance-before-no-return varies. The largest minimum distance is required when the MAV is approaching the wall at the angle $\beta = 90^\circ$. At any other angle the MAV can initiate the turning maneuver later since the amount of angle it needs to turn before colliding with the wall is smaller. So if one wants to use a single global threshold for initializing the turning maneuver one has to use the optical flow threshold for $\beta = 90^\circ$. Alternatively one can use a slightly more complex optical flow detection system and flight controller similarly to the one described by Zufferey et al. (2007) (section 4.1) where the optical flow measured by two sensors - one pointing to the left, another sensor pointing to the right of the MAV - are subtracted and the optical flow threshold and turning rate are adjusted based on the result of the subtraction. Since we are only interested in the highest bound of the $OF_{trans,max}/OF_{rot,max}$ -ratio it is sufficient to consider a single threshold for $OF_{trans,max}$ for β equal to 90° or $\pi/2$.

For the same reasons it is also sufficient to assume that the collision radius $R = 0$, and that the MAV always turns with its maximum rotation rate $\dot{\phi}$ which it can instantaneously reach from a rotation rate of zero.

The minimum time to rotate the MAV 90° or $\pi/2$ with the airplane's maximum rotation rate $\dot{\phi}$ can then be described as:

$$t_{rot} = \frac{\pi}{2\dot{\phi}} \quad (10.7)$$

Note that the translational optical flow is dependent on the distance, d_{nr} , in the direction that the sensor is pointing while the time to rotate depends on D_{nr} . The two distances are related by the angle α :

$$d_{nr} = \frac{D_{nr}}{\cos(\alpha)} \quad (10.8)$$

The translational optical flow measured at the distance-before-no-return, $OF_{trans,nr}$, can then be found by inserting (10.8) into (10.1) and setting $R = 0$:

$$OF_{trans,nr} = \frac{v}{d_{nr}} \cdot \sin(\alpha) \quad (10.9)$$

$$\begin{aligned} &= \frac{2v\dot{\phi} \sin(\alpha) \cos(\alpha)}{v\pi} \\ &= \frac{2\dot{\phi} \sin(\alpha) \cos(\alpha)}{\pi} \end{aligned} \quad (10.10)$$

For this analysis we are only interested in finding the maximal translational optical flow that needs to be detected by the optical flow sensor before turning away from the wall. Since the expression $\sin(\alpha) \cos(\alpha)$ reaches its maximum of 0.5 for an angle $\alpha = 45^\circ$ one can find the maximum optical flow caused by translation of an observer approaching a wall at the distance-before-no-return $OF_{trans,nr,max}$:

$$\begin{aligned} OF_{trans,nr,max} &= \frac{2\dot{\phi} \sin(45^\circ) \cos(45^\circ)}{\pi} \\ &= \frac{2\dot{\phi}}{2\pi} \\ &= \frac{\dot{\phi}}{\pi} \end{aligned} \quad (10.11)$$

and the ratio between the amounts of translational and rotational optical flow that need to be measured is

$$\frac{OF_{trans,max}}{OF_{rot,max}} = \frac{1}{\pi} \quad (10.12)$$

So given the rather conservative estimations used it can be concluded that **independent of the observer's velocity** the perceived maximum optical flow caused by the translation of the airplane that still needs to be detected before a turning maneuver has to be initiated is smaller than the optical flow caused by self-rotation of the airplane $OF_{\text{rot}} = \dot{\phi}$ **at least by a factor of π** .

10.2 Discussion

From the presented analysis it can be concluded that the extraction of translational optical flow by subtracting an estimation of the rotational optical flow from the total flow is problematic in the presence of noisy sensors if the translational optical flow component is a too small fraction of the total optical flow. If the approach should be applied successfully as presented by Zufferey et al. (2007) and others (chapter 4) the perceived rotational optical flow and thus the rotation rate of the MAV needs to be limited - reducing the maneuverability of the MAV. Even with this limitation of the rotation rate as presented in the wall-avoidance scenario, for constant-speed controlled MAVs the magnitude of the behaviorally relevant translational optical flow will always be smaller than the to be expected maximum optical flow generated during rotation of the MAV. An optical flow sensors that should measure the sum of both rotational and translational optical flow has to spend 2/3 of its resolution on the rotational component.

A flight controller that uses optical flow sensors more efficiently would be one that generates only translational optical flow and in case an obstacle needs to be avoided turns the MAV almost instantaneously. Such fast turns would not only allow a MAV to maneuver on very limited space. Furthermore since during the rapid turns it can be assumed that no behaviorally relevant event occurs, the optical flow sensors would only have to measure optical flow between turns when (almost) no rotational optical flow is generated. Thus the resolution of the optical flow sensors could be spend only on the translational optical flow leading to a better resolution of the relative distance to obstacles.

Many flies show such a flight behavior where periods of straight flights and rapid turns - called saccades - alternate (Collett and Land, 1975; Wagner, 1986; Schilstra and Hateren, 1999; Tammero and Dickinson, 2002a). Figure 10.3 is showing example trajectories of flies triggered by image expansions. Flies have been recorded by a overhead camera while flying in cylindric arenas with uniform (left) and textured (right) backgrounds. Tammero and Dickinson (2002a) showed that flies alter their course by about

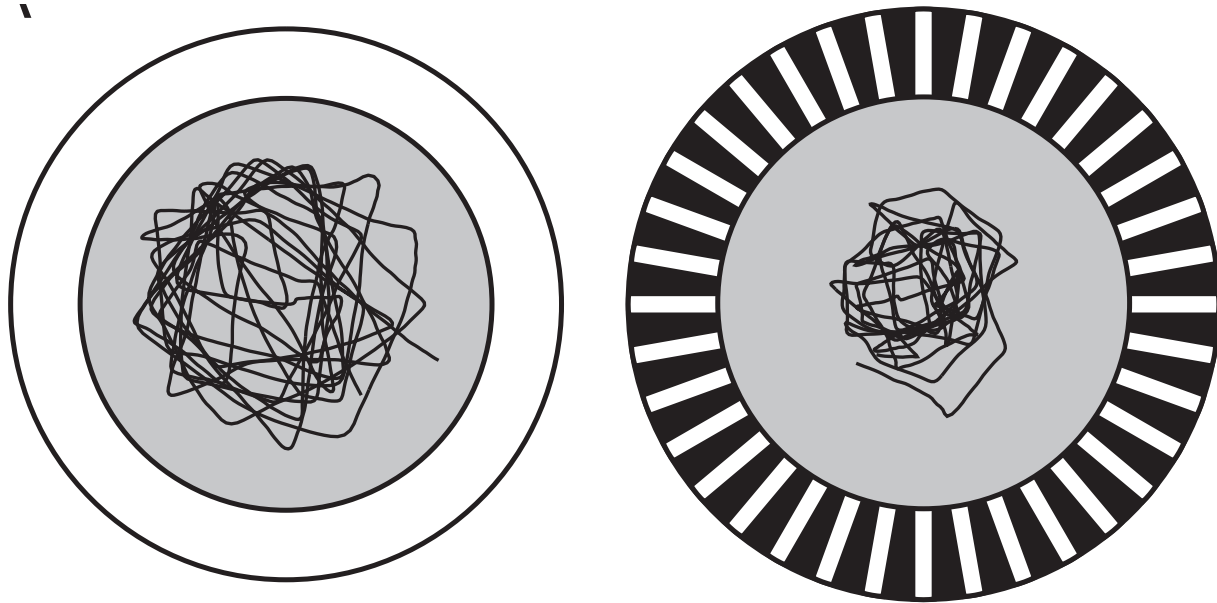


Figure 10.3: Example flight trajectories of flies flying in cylindric arenas with uniform (left) and textured (right) backgrounds. Flies show a behavior where periods of straight flights and rapid turns alternate. Turns are initiated by image expansions. The periods for straight flights are longer in the arena with uniform background and flies initiate turns closer to the walls. Figures are adapted from Tammero and Dickinson (2002a).

90° in less than 100ms. Saccades can lead to turns of more than 90°. Furthermore, Tammero and Dickinson (2002a) presented that once initiated, visual feedback does not appear to influence saccade kinematics further. So the flies seem indeed to measure optical flow only in between saccades.

Evidence has been found that while some of the saccades are spontaneously generated (Heide, 1983), other saccades seem to be triggered by image expansions (Tammero and Dickinson, 2002a). While saccades triggered by image expansions seem to be used to avoid obstacles, spontaneously generated saccades could be useful to observe the fly's environment at different angles in relation to the own movement. Also for MAVs spontaneous saccades could be helpful to resolve ambiguities in the perceived optical flow patterns and to maximize the perceived optical flow in different directions since as presented in section 2.3 (observation 4) zero optical flow is perceived in the center of an expansion pattern generated when flying into an obstacle but maximal optical flow is generated in all directions orthogonal to the flight direction. Spontaneous changes of the flight direction can thus help detecting obstacles that would otherwise generate to little optical flow when flying directly towards these obstacles. In insects spontaneous saccades might furthermore help confusing and detecting predators that try to chase a

flying insect by using motion camouflage.

Although optical flow sensors could be used more efficiently when MAV controllers would exploit saccades, similarly rapid turns as found in insects have not been demonstrated on state-of-the-art optical flow guided MAVs so far. Currently presented MAVs in the literature that have been successfully controlled with optical flow are either designed to be passively stable like the airplanes presented by Zufferey et al. (2007) and Beyeler et al. (2009) or critically stabilized helicopters and quadcopters (chapter 4). They are thus not capable of performing rapid saccades. Insects and bees however are passively unstable flyers (Taylor and Krapp, 2007) allowing for their impressive maneuverability and better use of their optical flow sensors. Quadcopters and helicopters when being controlled differently but especially flapping robots might be able to bridge the performance gap between insects and artificial MAVs in the future.

Chapter 11

Conclusion

11.1 Main achievements

This thesis contributes to two research areas: First to the area of VLSI optical flow sensors and second to the area of optical flow controlled micro aerial vehicles (MAVs). The thesis introduces novel VLSI circuits and bio-inspired optical flow sensors for steering autonomous MAVs. The thesis further aims at providing a better understanding of how MAVs can use optical flow sensors more efficiently and how the flight controller and flight properties of MAVs could be modified to allow the MAVs to mimic the flight behavior of insects more closely.

The main part of the thesis focuses on the development of custom Very Large Scale Integrated (VLSI) optic flow sensor chips that are suitable for autonomous micro aerial vehicles. Therefore both existing optical flow controlled MAVs as well as VLSI optical flow sensors are evaluated to specify the required properties of optical flow sensors suitable for autonomous micro aerial vehicles. The results of these studies are presented in chapter 4 and 6.

The on-board requirements for small, light-weight, low power sensors and electronics on autonomous micro aerial vehicles limit the computational power available for processing sensory signals including the computation of optic flow. Custom Very Large Scale Integrated (VLSI) sensor chips which perform both phototransduction as well as optic flow estimation on the focal plane can be beneficial for such platforms since they allow for specialized compact solutions that use little power while computing optic flow values in real-time. To be suitable for MAV control optical flow sensors must provide reliable output signals that ideally only depend on the velocity of the visual stimuli - not for instance on stimulus contrast and spatial frequency. Although such optical flow mea-

tures that are relatively unaffected by variations in the contrast and spatial frequency of the visual patterns have been found in insects (Si et al., 2003) the realization of this property in artificial systems is challenging.

This thesis proposes two VLSI sensor systems for controlling MAVs.

- The dynamic optical flow sensor (DOFS) described in chapter 7 implements a time-of-travel algorithm for computation of optical flow in form of custom VLSI circuits directly on the sensor's focal plane. It features a novel double threshold edge detection circuit that reliably generates output events for contrast values down to 2.5%. The sensor's sensitivity to contrast is thus improved by a factor greater than 7 in comparison to previously published sensors with similar architectures like the ones presented by Kramer et al. (1995, 1997). Due to the inclusion of the photoreceptor circuit by Delbrück and Mead (1994, 1995) and a modified version of the LMC circuit by Liu (2000), DOFS adapts over 4 decades of background intensity. Optical flow information can be extracted for visual stimulus velocities varying over two decades.
- The adaptive vision sensor (aVIS) is presented in chapter 9. In combination with the Image Interpolation Algorithm (I2A) by Srinivasan (1993, 1994) aVIS forms an optical flow sensor module for steering micro aerial vehicles. aVIS implements 160 pixels each containing adaptive photoreceptor circuits and LMC circuits Liu (2000). The LMC output signals are read by a microcontroller running the I2A that computes and integrates optical flow among aVIS pixels.

Table 11.1: Comparison of dynamic optical flow sensor (DOFS) and optical flow sensor module based on the adaptive vision sensor (aVIS).

Dynamic optical flow sensor	Adaptive vision sensor module
Token-based	Gradient-based
Event-based optical flow calculation / every event counts	Frame-based optical flow calculation / single calculations less important
(+) Optical flow calculation on-chip	(-) Optical flow calculation off-chip
Requires external spatial integration	(+) Built-in spatial integration due to Image Interpolation Algorithm

Table 11.1 compares key properties of the two presented optical flow sensor systems. Since both DOFS and aVIS rely on fundamentally different implementation principles there is no clear winner in a comparison. In DOFS the optical flow computation is only triggered when a contrast edge is detected. So if only few contrast are available within the scene each edge detection event becomes important and noisy calculations cannot be tolerated. For aVIS however a frame-based optical flow calculation algorithm was selected that requires sufficiently smooth contrast gradients to operate correctly. If sufficient gradients are present several optical flow calculations can be performed on a moving gradient where DOFS would only detect a single or very few contrast edges. Thus under these conditions the aVIS module can support temporal averaging among several measurements and single optical flow calculations - even if they are noisy - become less important. DOFS comes with the clear advantage that the entire optical flow computation process is implemented on-chip while aVIS requires an external microcontroller to perform the computation of optical flow. On the other hand if spatial integration of local optical flow measurements are required DOFS will require an external microcontroller as well. This can become computationally expensive for a DOFS implementations while spatial integration is a feature naturally embedded in the Image Interpolation Algorithm that was chosen to compute optical flow from the outputs of aVIS.

We believe that both DOFS and aVIS are well suited for behavioral guidance of MAVs like the airplane by Zufferey et al. (2007). A final demonstration of the sensor systems on MAVs is still missing but the sensors have been tested while generating optical flow on PC screen and by operating the sensors on rotating platforms. By comparing the sensors' outputs with output signals generated by rate gyroscope we can shown that the sensors are well suited for the flight controller proposed by Zufferey et al. (2007). Furthermore, DOFS has been demonstrated as part of a simple car driving simulation (chapter 8) where the outputs of DOFS are processed to steer a simulated car (Moeckel et al., 2008).

The application of the edge detection circuit presented in this thesis (chapter 7) is not restricted to its particular implementation as part of the dynamic optical flow sensors. Most token-based vision sensors require a reliable feature detection circuit that provides output events independent from stimulus contrast. Furthermore, the edge detection circuit allows for online adjustment of the amount of signal noise that should be rejected - a feature that is useful in many token-based vision sensors. So the presented edge detection circuit can be of benefit for token-based vision sensors in general.

From the evaluation of state-of-the-art optical flow controlled MAVs like the airplanes

by Zufferey et al. (2007) and Beyeler et al. (2009) presented in chapter 4 it can be concluded that the currently used MAVs are mostly chosen for their stable flight behavior. Insects however are passively unstable (Taylor and Krapp, 2007) and developed flight strategies that use optical flow sensors more efficiently than artificial flyers (chapter 10). As presented in chapter 3 insects try to keep the overall perceived optical flow constant (Srinivasan et al., 1996) rather than their forward velocity as it is currently the case in MAV control (chapter 4). Furthermore, insects like flies and honeybees developed a flight behavior where periods of straight flight alternate with fast turns - so called saccades (Schilstra and Hateren, 1999; Tammero and Dickinson, 2002a). This flight behavior uses optical flow sensors efficiently: During the straight flights only translational optical flow is generated that can be used for estimation of distances between the insect and objects in its environment. During saccades insects do not use optical flow for behavioral guidance (Tammero and Dickinson, 2002a). Since due to their body properties insects can perform fast saccades the time in which they do not estimate optical flow is not behavioral relevant. State-of-the-art optical flow controlled MAVs however typically cannot turn as fast as insects. Thus the standard MAV controller uses a gyroscope to estimate the distance independent optical flow component that is caused due to rotation. This rotational optical flow component is subtracted from the total flow measured by optical flow sensors to gain the distance dependent translational optical flow. In conclusion in contrast to flies and honeybees the state-of-the-art optical flow controlled MAVs do not only lack maneuverability but their optical flow sensors also have to measure both translational and rotational optical flow together and thus the sensors' resolution is not optimally used (chapter 10).

11.2 Future works

This thesis makes a contribution to the futuristic scenario where MAVs will show the same impressive flight behavior as flies or honeybees. But there is clearly still much work left to be done to reach this goal. Concrete future works for improving the optical flow sensors and MAVs are:

The event-based processing of optical flow in the presence of detected contrast edges opens the possibility to include an event-based communication interface to the dynamic optical flow sensor. Such a communication interfaces that only transmit data to a host processor in the event that new optical flow values are available would make the communication between the sensor and the host processor more efficient if it replaces the current scheme where the host processor is sequentially scanning and converting the

local optical flow outputs of the pixel array. Asynchronous communication interfaces with address-event representation (AER) have been widely studied for instance by Boahen (2000, 2004*a,b,c*). So far no AER interface was added to DOFS since it would as well require changes to the controller of the MAV.

So far the dynamic optical flow sensor currently does not include a powerful mechanism for the spatial integration of local optical flow outputs. If the pool of pixels for which a global optical flow estimate should be estimated could be selected online a on-chip mechanism for spatial integration would further ease the optical flow processing on the host processor.

The aVIS module implements spatial integration through the image interpolation algorithm (I2A). The experimental results presented in chapter 9 are encouraging and demonstrate that the I2A is well suited for processing optical flow based on the aVIS LMC output signals. In the future it would be worth transferring the I2A directly onto the adaptive vision sensor to free the microcontroller from the optical flow calculations.

Finally, from the presented evaluations it can be concluded that for closing the performance gap between insects and their engineered counterparts a change from passively stable airplanes to quadcopters or flappers that allow flight saccades would be useful.

List of Figures

2.1	Flying spherical camera and example of optical flow field.	12
2.2	Translation of a spherical sensor	14
2.3	Optical flow caused by translation of a spherical sensor.	15
2.4	Projection of a vector.	16
2.5	Optical flow caused by rotation of a spherical sensor.	17
2.6	Optical flow fields caused by translation of a spherical sensor and their Mercator maps.	19
2.7	Optical flow fields caused by rotation of a spherical sensor and their Mercator maps.	20
2.8	Time to contact a surface.	22
2.9	During translation of a sensor points can appear and disappear in the sensor's field of view.	23
3.1	Honeybees balance optical flow on right and left eye.	27
3.2	Honybees slow down in narrow tunnels.	28
3.3	Experiment proving that honeybees use optical flow to measure distances between the hive and food sources.	30
3.4	Picture and schematic of compound eye.	32
3.5	Interneurons in blowflies.	33
4.1	10-gram indoor airplane for wall avoidance (part 1)	41
4.2	10-gram indoor airplane for wall avoidance (part 2)	42
4.3	Airplane, optical mouse sensor and sensor array.	44
4.4	Visual field of optical mouse sensors.	45

4.5	Optical-flow controlled quadcopter.	47
5.1	Layout, structure and symbol of nFET and pFET.	55
5.2	3D schematic of nFET and pFET layout.	56
5.3	Characteristic curve of nFET for changing gate voltage.	58
5.4	Characteristic curve of nFET for changing drain voltage.	59
6.1	Image projection with thin lens.	74
6.2	Field of view.	75
6.3	Sensor layout for increased spatial resolution.	78
6.4	Timeline of single-chip VLSI optical flow sensor implementations.	84
6.5	Overview of motion algorithms implemented in aVLSI.	85
6.6	Comparison between time-of-travel motion algorithms.	92
7.1	Picture of DOFS with and without lens	103
7.2	Overview of dynamic optical flow sensor.	104
7.3	DOFS layout and picture	106
7.4	Optic flow output profile for the FS algorithm with linear decay rate.	109
7.6	Schematic and layout of DOFS photoreceptor circuit.	115
7.7	Schematics photoreceptor buffer circuit.	117
7.8	Schematic and layout of LMC circuit.	119
7.9	LMC circuit by Liu.	120
7.10	Current-voltage characteristics of adaptive resistive elements.	122
7.11	Step response of LMC circuit.	123
7.12	Schematic and layout of my novel edge detection circuit.	126
7.13	Timing diagram for edge detection circuit.	127
7.14	Output events from edge detection circuit for sinusoidal stimuli with varying frequencies.	129
7.15	Circuits for pulse shaping and generation of facilitate signal.	131
7.16	Sampling pulses for different stimulus frequencies.	132

7.17 Sample & hold & decay circuit.	134
7.18 Scanner circuit.	136
7.19 Schematic of scanner buffer.	137
7.20 Schematic of scanner shift register.	138
7.21 Direction selection circuit.	140
7.22 Measured outputs from the fabricated DOFS	141
7.23 Raw optical flow outputs.	142
7.24 Optic flow output profiles for individual pixels.	144
7.25 Optic flow output profiles for individual pixels.	145
7.26 Optic flow output profiles for individual pixels.	146
7.27 Optic flow output profiles for individual pixels.	147
7.28 Mean optical flow output profile.	148
7.29 Spatial distribution of optical flow output profiles.	149
7.30 Mean optical flow output profile.	150
7.31 Spatial distribution of optical flow output profiles.	151
7.32 Mean optical flow output profile (linearized).	152
7.33 Spatial distribution of linearized optical flow output profiles.	152
7.34 Mean optical flow output profile (linearized).	153
7.35 Spatial distribution of linearized optical flow output profiles.	153
7.36 Contrast dependence of optical flow outputs.	154
7.37 Comparison of optical flow outputs against rate gyroscope.	154
7.38 Self-biasing photoreceptor circuit.	156
7.39 Multi-decay rate facilitate circuit with output approximately linear to stimulus velocity.	159
7.40 Simulation plots of the multi-decay rate facilitate circuit.	160
8.1 Setup for car driving experiments with DOFS.	162
8.2 Screen shot of car driving simulation software.	163
8.3 Single-layer perceptron.	163

8.4	Perceptron learning results for DOFS controlled car driving simulation. . .	164
8.5	Car trajectory.	165
8.6	Screen shots of car driving simulation software with different contrast values.	166
9.1	Microphotograph of aVIS bare die.	171
9.2	Pictures of aVIS chip and sensor module.	172
9.3	Schematic and of aVIS photoreceptor circuit.	174
9.4	Schematics photoreceptor buffer circuit.	175
9.5	Schematic of aVIS LMC circuit.	176
9.6	Image frames for I2A optical flow calculation.	177
9.7	aVIS experimental room.	179
9.8	OF vs gyroscope raw data.	182
9.9	OF vs gyroscope raw data.	183
9.10	Timing diagram of OF and gyroscope.	185
9.11	Timing diagram of OF and gyroscope.	186
9.12	Timing diagram of OF and gyroscope.	187
9.13	Timing diagram of OF and gyroscope.	188
9.14	Timing diagram of OF and gyroscope.	189
9.15	Timing diagram of OF and gyroscope.	190
9.16	Optic flow versus gyroscope output.	191
9.17	Optic flow versus gyroscope output.	192
9.18	Optic flow versus gyroscope output.	193
9.19	Optic flow versus gyroscope output.	194
9.20	Optic flow versus gyroscope output.	195
9.21	Optic flow versus gyroscope output.	196
9.22	TSL results.	197
9.23	Summary of RMSE measurements.	198
10.1	Wall avoidance with an optical flow guided MAV.	200

10.2 Distance dependence of translational optical flow.	202
10.3 Flight saccades in flies.	206

List of Tables

- 4.1 Comparison of properties of optical-flow controlled MAVs presented in this section. 49

- 5.1 Regimes of transistor operation: nFET 60
- 5.2 Regimes of transistor operation: pFET 60
- 5.3 Capacitance of poly1-to-poly2 and MOS capacitors. 62

- 7.1 Specification of dynamic optical flow sensor and lens. 102

- 9.1 aVIS and lens specification. 170
- 9.2 Specifications of TSL3301 image sensor. 181

- 11.1 Comparison of dynamic optical flow sensor (DOFS) and optical flow sensor module based on the adaptive vision sensor (aVIS). 210

Index

aperture problem, 80

aVLSI, 53

CMOS, 56

depth of field, 75

dummy device, 66

f-number, 75

Facilitate-and-sample, 94

FET, 54

field of view, 75

fixed-pattern noise, 64

floating gate, 68

focal length, 74

FPN, 64

FS, 94

image motion, 11

MOSFET, 54

motion segmentation, 81

nFET, 56

optical flow, 11

optical flow sensors, 12

pFET, 56

phototransduction, 5

System-on-chip, 67

transistor, 54

Bibliography

- Adelson, E. and Bergen, J. (1985). Spatio-temporal energy models for the perception of motion. *Journal of the Optical Society of America A*, **2**, 284–299.
- Alspector, J. and Allen, R. B. (1987). A neuromorphic VLSI learning system. *in* R. Losleben, ed., *Proceedings of the Stanford Conference on Advanced Research in VLSI*. MIT Press. pp. 313–349.
- Andreou, A., Strohbehn, K. and Jenkins, R. (1991). Silicon retina for motion computation. *IEEE International Symposium on Circuits and Systems*, pp. 1373–1376.
- Arreguit, X., Van Schaik, F. A., Bauduin, F., Bidiville, M. and Raeber, E. (1996). A CMOS motion detector system for pointing devices. *IEEE Journal of Solid-State Circuits*, **31**(12), 1916–1921.
- Aslam-Siddiqi, A., Brockherde, W., Schanz, M. and Hosticka, B. J. (1998). A 128-pixel CMOS image sensor with integrated nonvolatile memory. *IEEE Journal of Solid-State Circuits*, **33**(10), 1497–1501.
- Banu, M. and Tsividis, Y. (1982). Floating voltage-controlled resistors in CMOS technology. *Electronic Letters*, **18**(15), 678–679.
- Barber, B., Griffiths, S., McLain, T. and Beard, R. (2005). Autonomous landing of miniature aerial vehicles. *AIAA Infotech@Aerospace*, .
- Barlow, H. B. and Levick, W. R. (1965). The mechanism of directionally selective units in rabbit's retina. *Journal of Physiology*, **178**, 477–504.
- Bartz, R. (2011). *Volucella pellucens* is a hover-fly. Available online at http://de.wikipedia.org/w/index.php?title=Datei:Volucella_pellucens_head_complete_Richard_Bartz.jpg&filetimestamp=20070827134605; visited on December 28th 2011.

- Bastos, J., Steyaert, M., Pergoot, A. and Sansen, W. (1997). Mismatch characterization of submicron MOS transistors. *Analog Integrated Circuits and Signal Processing*, **12**, 95–106.
- Benson, R. and Delbruck, T. (1992). Direction selective silicon retina that uses null inhibition. *Advances in Neural Information Processing Systems*, **4**, 756–763.
- Beyeler, A., Zufferey, J. and Floreano, D. (2009). Vision-based control of near-obstacle flight. *Autonomous Robots*, **27**(3), 201–219.
- Blanes, C. (1986). Appareil visuel élémentaire pour la navigation à vue d'un robot mobile autonome. Master thesis in neuroscience. Univ. of Aix-Marseille II. Marseille, France.
- Boahen, K. (2000). Point-to-point connectivity between neuromorphic chips using address events. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, **47**(5), 416–434.
- Boahen, K. (2004a). A burst-mode word-serial address-event link-I: Transmitter design. *IEEE Transactions on Circuits and Systems I: Regular Papers*, **51**(7), 1269–1280.
- Boahen, K. (2004b). A burst-mode word-serial address-event link-II: Receiver design. *IEEE Transactions on Circuits and Systems I: Regular Papers*, **51**(7), 1281–1291.
- Boahen, K. (2004c). A burst-mode word-serial address-event link-III: Analysis and test results. *IEEE Transactions on Circuits and Systems I: Regular Papers*, **51**(7), 1292–1300.
- Boch, R. (1957). Rassenmässige Unterschiede bei den Tänzen der Honigbiene (*Apis mellifica* L.). *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, **40**(3), 289–320.
- Bouguet, J. (2001). Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. Technical report.
- Brainard, D. H. (1997). The psychophysics toolbox. *Spatial Vision*, **10**, 433–436.
- Burton, A. and Radford, J. (1978). *Thinking in Perspective: Critical Essays in the Study of Thought Processes*. Routledge.
- Chahl, J., Srinivasan, M. and Zhang, S. (2004). Landing strategies in honeybees and applications to uninhabited airborne vehicles. *The International Journal of Robotics Research*, **23**(2), 101.

- Collett, T. and Land, M. (1975). Visual control of flight behaviour in the hoverfly *Syrphoctonus pipiens* L. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, **99**(1), 1–66.
- David, C. T. (1982). Compensation for height in the control of ground speed by *Drosophila* in a new, 'barber's pole' wind tunnel. *Journal of Computational Physiology A*, **147**, 485–493.
- Delbruck, T. (1993). Silicon retina with correlation-based, velocity tuned pixels. *IEEE Transactions on Neural Networks*, **4**(3), 529–541.
- Delbruck, T. and Lichtsteiner, P. (2006). Fully programmable bias current generator with 24 bit resolution per bias. *IEEE International Symposium on Circuits and Systems*, pp. 2849–2852.
- Delbrück, T. and Mead, C. (1994). Analog VLSI adaptive logarithmic wide-dynamic-range photoreceptor. *in* IEEE International Symposium on Circuits and Systems. Vol. 94. pp. 339–342.
- Delbrück, T. and Mead, C. (1995). Analog VLSI phototransduction by continuous-time, adaptive, logarithmic photoreceptor circuits. *in* C. Koch and H. Li, eds, *Vision Chips: Implementing vision algorithms with analog VLSI circuits*. IEEE Computer Society Press. pp. 139–161.
- Delbruck, T. and Oberhoff, D. (2004). Self-biasing low power adaptive photoreceptor. *in* International Symposium on Circuits and Systems. Vol. 4. IEEE. pp. IV–844.
- Delbrueck, T. and Mead, C. (1989). An electronic photoreceptor sensitive to small changes in intensity. *Advances in neural information processing systems*, **1**, 720–727.
- Deutschmann, R. A. and Koch, C. (1998a). Compact analog VLSI 2-D velocity sensor. *IEEE International Conference on Intelligent Vehicles*, **1**, 359–364.
- Deutschmann, R. A. and Koch, C. (1998b). Compact real-time 2-D gradient based analog VLSI motion sensor. *Proceedings of SPIE International Conference on Advanced Focal Plan Array and Electronic Cameras II*, **3410**, 98–108.
- Dickinson, M. (1999). Haltere-mediated equilibrium reflexes of the fruit fly, *Drosophila melanogaster*. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, **354**(1385), 903–916.

- Diorio, C., Hasler, P., Minch, B. A. and Mead, C. (1997). Floating-gate MOS learning array with locally computed weight updates. *IEEE Transactions on Electron Devices*, **44**(12), 2281–2289.
- Egelhaaf, M. and Borst, A. (1993a). Motion computation and visual orientation in flies. *Comp Biochem Physiol*, **104A**, 659–673.
- Egelhaaf, M. and Borst, A. (1993b). Movement detection in arthropods. *Visual motion and its role in the stabilization of gaze*, pp. 53–77.
- Enz, C. C. and Temes, G. C. (1996). Circuit techniques for reducing the effects of op-amp imperfections: autozeroing, correlated double sampling, and chopper stabilization. *Proceedings of the IEEE*, **84**(11), 1584–1614.
- Esch, H. and Burns, J. (1995). Honeybees use optic flow to measure the distance of a food source. *Naturwissenschaften*, **82**(1), 38–40.
- Esch, H. and Burns, J. E. (1996). Distance estimation by foraging honeybees. *Journal of Experimental Biology*, **199**, 155–162.
- Esch, H., Zhang, S., Srinivasan, M. and Tautz, J. (2001). Honeybee dances communicate distances measured by optic flow. *Nature*, **411**(6837), 581–583.
- Etienne-Cummings, R., Fernando, S., Takahashi, N., Shtonov, V., Van der Spiegel, J. and Mueller, P. (1993). A new temporal domain optical flow measurement technique for focal plane VLSI implementation. *IEEE Proceedings on Computer Architectures for Machine Perception*, pp. 241–250.
- Figueroa, M., Bridges, S. and Diorio, C. (2004). On-chip compensation of device mismatch effects in analog VLSI neural networks. *Neural Information Processing Systems*, .
- Figueroa, M., Hyde, J., Hymes, T. and Diorio, C. (2001). A floating-gate trimmable high-resolution DAC in standard 0.25 μm CMOS. *Proceedings of the IEEE Nonvolatile Semiconductor Memory Workshop*, pp. 46–47.
- Forti, F. Wright, M. E. (1994). Measurement of MOS current mismatch in the weak inversion region. *IEEE Journal of Solid-State Circuits*, **29**(2), 138–142.
- Franceschini, N., Blanes, C. and Oufar, L. (1986). Appareil de mesure passif et sans contact de la vitesse d'un objet quelconque. Dossier technique Nb 51549. AN-VAR/DVAR. Paris.

- Fry, P. W., Noble, P. J. W. and Rycroft, R. J. (1970). Fixed-pattern noise in photomatrices. *IEEE Journal of Solid State Circuits*, **SC-5**(5), 250–254.
- Fry, S. N., Rohrseitz, N., Straw, A. D. and Dickinson, M. H. (2009). Visual control of flight speed in *drosophila melanogaster*. *Journal of Experimental Biology*, **212**, 1120–1130.
- Gibson, J. (1950). *The perception of the visual world*. Houghton Mifflin. Boston.
- Gottardi, M. and Yang, W. (1993). A CCD/CMOS image motion sensor. *International Solid State Circuits Conference*, pp. 194–195.
- Green, W. E., Oh, P. Y. and Barrows, G. (2004). Flying insect inspired vision for autonomous aerial robot maneuvers in near-earth environments. *in Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*. Vol. 3. IEEE. pp. 2347–2352.
- Green, W., Oh, P., Sevcik, K. and Barrows, G. (2003). Autonomous landing for indoor flying robots using optic flow. *in ASME International Mechanical Engineering Congress and Exposition*. Citeseer.
- Griffiths, S., Saunders, J., Curtis, A., Barber, B., McLain, T. and Beard, R. (2007). Obstacle and terrain avoidance for miniature aerial vehicles. *Advances in Unmanned Aerial Vehicles*, pp. 213–244.
- Gruev, V. and Etienne-Cummings, R. (2004). Active pixel sensor with on-chip normal flow computation on the read out. *IEEE International Conference on Electronics, Circuits, and Systems*, pp. 215–218.
- Han, I. and Park, S. B. (1984). Voltage-controlled linear resistor by two MOS transistors and its application to active RC filter MOS integration. *Proceedings of the IEEE*, **72**(11), 1655–1657.
- Han, S.-W. and Yoon, E. (2006). Area-efficient correlated double sampling scheme with single sampling capacitor for CMOS image sensors. *Electronic Letters*, **42**(6), 335–337.
- Harrison, R. and Koch, C. (1998). An analog VLSI model of the fly elementary motion detector. *Advances in neural information processing systems*, pp. 880–886.
- Harrison, R. and Koch, C. (1999). A robust analog VLSI motion sensor based on the visual system of the fly. *Autonomous Robots*, **7**(3), 211–224.

- Harrison, R. R. (2005). A biologically-inspired analog IC for visual collision detection. *IEEE Transactions on Circuits and Systems - I*, **52**, 2308–2318.
- Hasler, P., Minch, A. and Diorio (2001). An autozeroing floating-gate amplifier. *IEEE Transactions on Circuits and Systems II*, **48**(1), 74–82.
- Hassenstein, B. and Reichardt, W. (1956). Systemtheoretische Analyse der Zeit-, Reihenfolgen- und Vorzeichenauswertung bei der Bewegungsperzeption des Rüsselkäfers *Chlorophanus*. *Zeitschrift für Naturforschung*, **11b**, 513–524.
- Hausen, K. (1993). Decoding of retinal image flow in insects.. *Reviews of oculomotor research*, **5**, 203.
- Hausen, K. and Egelhaaf, M. (1989). Neural mechanisms of visual course control in insects. , .
- Heide, G. (1983). Neural mechanisms of flight control in Diptera. in W. Nachtigall, ed., *Insect Flight II*, Biona Report 2. G. Fischer. Stuttgart. pp. 35–52.
- Helmholtz, H. (1925). Physiological optics. *Optical Society of America*, **3**.
- Hengstenberg, R. (1982). Common visual response properties of giant vertical cells in the lobula plate of the blowflycalliphora. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, **149**(2), 179–193.
- Hengstenberg, R. (1993). Multisensory control in insect oculomotor systems. *Rev Oculomot Res*, **5**, 285–298.
- Hengstenberg, R., Hausen, K. and Hengstenberg, B. (1982). The number and structure of giant vertical cells (vs) in the lobula plate of the blowflycalliphora erythrocephala. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, **149**(2), 163–177.
- Heran, H. (1956). Ein Beitrag zur Frage nach der Wahrnehmungsgrundlage der Entfernungsweisung der Bienen (*Apis mellifica* L.). *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, **38**(1-2), 168–218.
- Higgins, C. and Korrapti, S. (2000). An analog VLSI motion energy sensor based on the Adelson-Bergen algorithm. *International ICSC Symposium on Biologically-Inspired Systems*, .

- Higgins, C. M., Pant, V. and Deutschmann, R. (2005). Analog VLSI implementation of spatio-temporal frequency tuned visual motion algorithms. *IEEE Transactions on Circuits and Systems - I*, **52**(3), 489–502.
- Hofstein, S. R. and Heiman, F. P. (1963). The silicon insulated gate field effect transistor. *Proceedings of the IEEE*, **51**.
- Horiuchi, T., Lazzaro, J., Moore, A. and Koch, C. (1991). A delay line based motion detection chip. *Advances in Neural Information Processing Systems*, **3**, 406–412.
- Horn, B. and Schunk, B. (1981). Determining optical flow. *Artificial Intelligence*, **17**, 185–203.
- Horridge, G. (1987). The evolution of visual processing and the construction of seeing systems. *Proceedings of the Royal society of London. Series B. Biological sciences*, **230**(1260), 279.
- Kahng, D. (1963). Electric field controlled semiconductor device. *U.S. Patent No. 3,102,230*, .
- Koenderink, J. J. and van Doorn, A. J. (1987). Facts on optic flow. *Biological Cybernetics*, **56**, 247–254.
- Kramer, J. (1996). Compact integrated motion sensor with three-pixel interaction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**, 455–460.
- Kramer, J., Sarpeshkar, R. and Koch, C. (1995). An analog VLSI velocity sensor. *IEEE International Symposium on Circuits and Systems*, pp. 413–416.
- Kramer, J., Sarpeshkar, R. and Koch, C. (1997). Pulse-based analog VLSI velocity sensors. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, **44** (2), 86–101.
- Krapp, H. (1999). Neuronal matched filters for optic flow processing in flying insects. *International review of neurobiology*, **44**, 93–120.
- Krapp, H. G. and Hengstenberg, R. (1996). Estimation of self-motion by optic flow processing in single visual interneurons. *Nature*, **384**, 463–466.
- Krapp, H., Hengstenberg, B. and Hengstenberg, R. (1998). Dendritic structure and receptive-field organization of optic flow processing interneurons in the fly. *Journal of Neurophysiology*, **79**(4), 1902.

- Lakshmikumar, K. R., Hadaway, R. A. and Copeland, M. A. (1986). Characterization and modeling of mismatch in MOS transistors for precision analog design. *IEEE Journal of Solid-State Circuits*, **21**(6), 1057–1066.
- Lenzinger, M. and Snow, E. H. (1969). Fowler-Nordheim tunneling into thermally grown SiO₂. *Journal of Applied Physics*, **40**(1), 278–283.
- Leven, S., Zufferey, J. and Floreano, D. (2009). A minimalist control strategy for small uavs. *in Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on. IEEE.* pp. 2873–2878.
- Lichtsteiner, P., Posch, C. and Delbruck, T. (2006). A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change. *in IEEE International Solid-State Circuits Conference. IEEE.* pp. 2060–2069.
- Lichtsteiner, P., Posch, C. and Delbruck, T. (2008). A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits*, **43**(2), 566–576.
- Lilienfeld, J. E. (1930). Method and apparatus for controlling electric currents. *U.S. Patent No. 1,754,145*, .
- LIS (2011). Compound eye. Available online at Laboratory of Intelligent Systems, EPFL <http://lis.epfl.ch/curvace>; visited on December 28th 2011.
- Liu, S.-C. (1999). Silicon photoreceptors with controllable adaptive filtering properties. *in C. G and M. Bayoumi, eds, Learning on Silicon. Kluwer Academic Publishers. Norwell, MA.* pp. 67–78.
- Liu, S.-C. (2000). A neuromorphic aVLSI model of global motion processing in the fly. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, **47**(12), 1458–1467.
- Liu, S.-C., Kramer, J., Indiveri, G., Delbruck, T. and Douglas, R. (2002). *Analog VLSI: Circuits and Principles*. MIT Press. Boston.
- Liu, S.-C. and Moeckel, R. (2008). Temporally learning floating-gate VLSI synapses. *IEEE International Symposium on Circuits and Systems*, pp. 2154–2157.
- Lukas, B. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. *in Proceedings of the 7th International Joint Conference on Artificial Intelligence.* pp. 674–679.

- Lyon, R. (1981). The optical mouse, and an architectural methodology for smart digital sensors. *CMU Conference on VLSI Structures and Computations*, pp. 1–19.
- Mead, C. A. and Delbruck, T. (1991). Scanners for visualizing activity of analog VLSI circuitry. *Analog Integrated Circuits and Signal Processing*, **1**, 93–106.
- Mehta, S. and Etienne-Cummings, R. (2003). Normal optical flow chip. *IEEE International Symposium on Circuits and Systems*, **4**, 784–787.
- Mehta, S. and Etienne-Cummings, R. (2004). Normal optical flow measurement on a CMOS APS imager. *IEEE International Symposium on Circuits and Systems*, **4**, 848–851.
- Mehta, S. and Etienne-Cummings, R. (2006). A simplified normal optical flow CMOS camera. *IEEE Transactions on Circuits and Systems - I*, **53**(6), 1223–1234.
- Meitzler, R. C., Strohbehn, K. and Andreou, A. G. (1995). A silicon retina for 2-D position and motion computation. *IEEE International Symposium on Circuits and Systems*, **III**, 2096–2099.
- Miller, K. and Borrows, G. (1999). Feature tracking linear optic flow sensor chip. *in Circuits and Systems, 1999. ISCAS'99. Proceedings of the 1999 IEEE International Symposium on*. Vol. 5. IEEE. pp. 116–119.
- Minch, B. A., Diorio, C., Hasler, P. and Mead, C. (1996). The matching of small capacitors for analog VLSI. *IEEE International Symposium on Circuits and Systems*, **1**, 239–241.
- Moeckel, R. (2005). Development of a STDP floating-gate synapse in CMOS VLSI. Master's thesis. University of Rostock. Germany.
- Moeckel, R., Jaeggi, R. and Liu, S. (2008). Steering with an avlsi motion detection chip. *in IEEE International Symposium on Circuits and Systems*. IEEE. pp. 1036–1039.
- Moeckel, R. and Liu, S.-C. (2007). Motion detection circuits for a time-to-travel algorithm. *IEEE International Symposium on Circuits and Systems*, pp. 3079–3082.
- Moeckel, R. and Liu, S.-C. (2008). A time-of-travel motion detection chip for steering autonomous robots. *Workshop on Visual guidance systems for small autonomous aerial vehicles at IEEE/RSJ International Conference on Intelligent Robots and Systems*, .

- Moeckel, R. and Liu, S.-C. (2009). Motion detection chips for robotic platforms. *in* D. Floreano, J. Zufferey, M. Srinivasan and C. Ellington, eds, *Flying Insects and Robots*. Springer.
- Nagaraj, K. (1986). New CMOS floating voltage-controlled resistor. *Electronic Letters*, **22**(12), 667–668.
- Nalbach, G. (1993). The halteres of the blowfly calliphora. 1. Kinematics and dynamics.. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, **173**(3), 293–300.
- Nalbach, G. and Hengstenberg, R. (1994). The halteres of the blowfly calliphora. 2. Three-dimensional organization of compensatory reactions to real and simulated rotations.. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, **175**(6), 695–708.
- Nay, K. and Budak, A. (1983). A voltage-controlled resistance with wide dynamic range and low distortion. *IEEE Transactions on Circuits and Systems*, **30**(10), 770–772.
- Netter, T. and Francheschini, N. (2002). A robotic aircraft that follows terrain using a neuromorphic eye. *in* *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*. Vol. 1. IEEE. pp. 129–134.
- Nicoud, J. and Zufferey, J. (2002). Toward indoor flying robots. *in* *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*. Vol. 1. IEEE. pp. 787–792.
- Osa, J. I., Porta, S. and Carlosena, A. (1998). The most resistive model for the MOS resistive circuit. *IEEE International Symposium on Circuits and Systems*, **2**, 208–211.
- Pant, V. and Higgins, C. M. (2007). A biomimetic focal plane speed computation architecture. *Proceedings of the Computational Optical Sensing and Imaging Conference*, .
- Papazoglou, C. A. and Karybakas, C. A. (1999). Electronically tunable floating CMOS resistor independent of the MOS parameters and temperature. *IEEE International Conference on Circuits and Systems*, **1**, 311–314.
- Pavasovic, A., Andreou, A. G. and Westgate, C. R. (1994). Characterization of sub-threshold MOS mismatch in transistors for VLSI systems. *Analog Integrated Circuits and Signal Processing*, **6**, 75–85.
- Pelgrom, M. J., Duinmaijer, A. C. J. and Welbers, A. P. G. (1989). Matching properties of MOS transistors. *IEEE Journal of Solid-State Circuits*, **SC-24**(5), 1433–1440.

- Pelli, D. G. (1997). The videotoolbox software for visual psychophysics: Transforming numbers into movies. *Spatial vision*, **10**(4), 437–442.
- Popa, C. (2007). Linearized CMOS active resistor independent on the bulk effect. *Proceedings of the 17th Great Lakes Symposium on VLSI*, pp. 253–256.
- Ramirez-Angulo, J., Sawant, M. S., Carvajal, R. G. and Lopez-Martin, A. (2005). Linearization of MOS resistors using capacitive gate voltage averaging. *Electronic Letters*, **41**(9), 511–512.
- Reichardt, W. (1961). Autocorrelation, a principle for the evaluation of sensory information by the central nervous system. *Sensory Communication*, pp. 303–317.
- Reichardt, W. (1987). Evaluation of optical motion information by movement detectors. *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, **161**(4), 533–547.
- Reichel, L., Liechti, D., Presser, K. and Liu, S.-C. (2005). Range estimation on a robot using neuromorphic motion sensors. *Robotics and Autonomous Systems*, **51**(2-3), 167–174.
- Rodriguez, A. F., Andersen, E., Bradley, J. M. and Taylor, C. N. (2007). Wind estimation using an optical flow sensor on a miniature air vehicle. *AIAA Infotech@Aerospace*, .
- Rohrseitz, N. and Fry, S. (2011). Behavioural system identification of visual flight speed control in *Drosophila melanogaster*. *Journal of The Royal Society Interface*, **8**(55), 171.
- Rossel, S. (1983). Binocular stereopsis in an insect. *Nature*, **302**, 821–822.
- Ruffier, F. and Franceschini, N. (2004). Visually guided micro-aerial vehicle: automatic take off, terrain following, landing and wind reaction. *in Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on. Vol. 3. IEEE.* pp. 2339–2346.
- Ruffier, F. and Franceschini, N. (2005). Optic flow regulation: the key to aircraft automatic guidance. *Robotics and Autonomous Systems*, **50**(4), 177–194.
- Ruffier, F. and Franceschini, N. (2008). Aerial robot piloted in steep relief by optic flow sensors. *in Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on. IEEE.* pp. 1266–1273.

- Sah, C. T. (1961). A new semiconductor triode, the surface-potential controlled transistor. *Proceedings of the IRE*, **49**(11), 1623–1634.
- Sakurai, S. and Ismail, M. (1993). A CMOS square-law programmable floating resistor. *IEEE International Symposium on Circuits and Systems*, **39**(8), 1184–1187.
- Sanchez, J. J. and DeMassa, T. A. (1991). Review of carrier injection in the silicon/silicon-dioxide system. *IEEE Proceedings G: Circuits, Devices and Systems*, **138**(3), 377–389.
- Santen, v. J. and Sperling, G. (1984). A temporal covariance model of motion perception. *Journal of the Optical Society of America A*, **1**, 451–473.
- Sarpeshkar, R., Bair, W. and Koch, C. (1993). Visual motion computation in analog VLSI using pulses. *Advances in Neural Information Processing Systems*, **5**, 781–788.
- Scaramuzza, D., Martinelli, A. and Siegwart, R. (2006). A toolbox for easy calibrating omnidirectional cameras. *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, .
- Schiller, P. H., Finlay, B. L. and Volman, S. F. (1976). Qualitative studies of single-cell properties in monkey striate cortex. I. Spatiotemporal organization of receptive fields. *Journal of Neurophysiology*, **39**, 1288–1319.
- Schilstra, C. and Hateren, J. (1999). Blowfly flight and optic flow. i. thorax kinematics and flight dynamics. *Journal of experimental biology*, **202**(11), 1481.
- Sellami, L. (1997). Linear bilateral CMOS resistor for neural-type circuits. *IEEE 40th Midwest Symposium on Circuits and Systems*, **2**, 1330–1333.
- Serrano-Gotarredona, T. and Linares-Barranco, B. (1999a). A 5-parameter mismatch model for short channel MOS transistors. *Proceedings of the 1999 European Solid State Circuits Conference*, pp. 440–443.
- Serrano-Gotarredona, T. and Linares-Barranco, B. (1999b). Systematic width-and-length dependent CMOS transistor mismatch characterization and simulation. *Analog Integrated Circuits and Signal Processing, Kluwer Academic Publishers*, **21**, 271–296.
- Serrano-Gotarredona, T. and Linares-Barranco, B. (2000). A new 5-parameter MOS transistor mismatch model. *IEEE Electron Device Letters*, **21**(1), 37–39.

- Sherman, A. and Dickinson, M. (2003). A comparison of visual and haltere-mediated equilibrium reflexes in the fruit fly *Drosophila melanogaster*. *Journal of experimental biology*, **206**(2), 295.
- Shi, J. and Tomasi, C. (1994). Good features to track. in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*. IEEE. pp. 593–600.
- Shyu, J. B., Temes, G. C. and Krummenacher, F. (1984). Random error effects in matched MOS capacitors and current sources. *IEEE Journal of Solid-State Circuits*, **SC-19**(6), 948–956.
- Shyu, J. B., Temes, G. C. and Yao, K. (1982). Random errors in MOS capacitors. *IEEE Journal of Solid-State Circuits*, **SC-17**(6), 1070–1076.
- Si, A., Srinivasan, M. and Zhang, S. (2003). Honeybee navigation: properties of the visually driven odometer'. *Journal of experimental biology*, **206**(8), 1265.
- Singh, S. P., Hansom, J. V. and Vlach, J. (1989). A new floating resistor for CMOS technology. *IEEE Transactions on Circuits and Systems*, **36**(9), 1217–1220.
- Soccol, D., Thurrowgood, S. and Srinivasan, M. (2007). A vision system for optic-flow-based guidance of uavs. in *Proceedings, Ninth Australasian Conference on Robotics and Automation, Brisbane*. pp. 10–12.
- Srinivasan, M., Lehrer, M., Kirchner, W. and Zhang, S. (1991). Range perception through apparent image speed in freely flying honeybees. *Visual neuroscience*, **6**(05), 519–535.
- Srinivasan, M., Thurrowgood, S. and Soccol, D. (2009). From visual guidance in flying insects to autonomous aerial vehicles. in D. Floreano, J. Zufferey, M. Srinivasan and C. Ellington, eds, *Flying Insects and Robots*. Springer.
- Srinivasan, M., Thurrowgood, S. and Soccol, D. (2011). Mav guidance inspired by principles of insect vision. *Encyclopedia of Aerospace Engineering*, .
- Srinivasan, M. V. (1993). An image-interpolation technique for the computation of 2-D motion. *Australian and New Zealand Conference on Intelligent Information Processing Systems*, pp. 367–371.
- Srinivasan, M. V. (1994). An image-interpolation technique for the computation of optic flow and egomotion. *Biological Cybernetics*, **71**, 401–416.

- Srinivasan, M. V. and Zhang, S. (2004). Visual motor computations in insects. *Annual Review of Neuroscience*, **27**, 679–696.
- Srinivasan, M. V., Zhang, S. W., Altwein, M. and Tautz, J. (2000a). Honeybee navigation: nature and calibration of the 'odometer'. *Science*, **287**, 851–853.
- Srinivasan, M. V., Zhang, S. W., Chahl, J. S., Barth, E. and Venkatesh, S. (2000b). How honeybees make grazing landings on flat surfaces. *Biological Cybernetics*, **83**, 171–183.
- Srinivasan, M., Zhang, S. and Bidwell, N. (1997). Visually mediated odometry in honeybees. *Journal of Experimental Biology*, **200**(19), 2513.
- Srinivasan, M., Zhang, S., Lehrer, M. and Collett, T. (1996). Honeybee navigation en route to the goal: visual flight control and odometry. *Journal of Experimental Biology*, **199**(1), 237–244.
- Stocker, A. A. (2002). An improved 2D optical flow sensor for motion segmentation. *IEEE International Symposium on Circuits and Systems*, **2**, 332–335.
- Stocker, A. and Douglas, R. (1999). Computation of smooth optical flow in a feedback connected analog network. *Advances in Neural Information Processing Systems*, **11**, 706–712.
- Takagi, S., Czarnul, Z. and Fuji, N. (1994). A synthesis of highly linear MOS circuits and their application to filter realization. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, **E77-A**(2), 351–355.
- Tammero, L. and Dickinson, M. (2002a). The influence of visual landscape on the free flight behavior of the fruit fly *Drosophila melanogaster*. *Journal of Experimental Biology*, **205**(3), 327.
- Tammero, L. F. and Dickinson, M. H. (2002b). Collision avoidance and landing responses are mediated by separate pathways in the fruit fly, *Drosophila melanogaster*. *Journal of Experimental Biology*, **205**, 2785–2798.
- Tanner, J. and Mead, C. (1986). An integrated analog optical motion sensor. *VLSI Signal Processing*, **2**, 59–76.
- Taylor, G. K. and Krapp, H. G. (2007). Sensory systems and flight stability: What do insects measure and why?. in J. Casas and S. Simpson, eds, *Insect Mechanics and Control*. Vol. 34 of *Advances in Insect Physiology*. Academic Press. pp. 231 – 316.

- Tran, P. T. and Wilamowski, B. M. (2001). VLSI implementation of cross-coupled MOS resistor circuits. *Conference of the IEEE Industrial Electronics Society*, pp. 1886–1891.
- Tsividis, Y. (1986). Continuous-time MOSFET-C filters in VLSI. *IEEE Transactions on Circuits and Systems*, **33**(2), 125–134.
- Ullman, S. (1981). Analysis of visual motion by biological and computer systems. *IEEE Computer*, **14**, 57–69.
- Vavelidis, K. and Tsividis, Y. (1992). Linear electronically tunable resistor. *Electronic Letters*, **28**(6), 2303–2305.
- Vavelidis, K. and Tsividis, Y. (1993). Design considerations for a highly linear electronically tunable resistor. *IEEE International Symposium on Circuits and Systems*, **2**, 1180–1183.
- Vavelidis, K., Tsividis, Y., Eynde, F. O. and Papananos, Y. (1997). Six-terminal MOS-FETs: Modeling and applications in highly linear, electronically tunable resistors. *IEEE Journal of Solid-State Circuits*, **32**(1), 4–12.
- Von Frisch, K. (1967). The dance language and orientation of bees. , .
- Wagner, H. (1986). Flight performance and visual control of flight of the free-flying housefly (*Musca domestica* L.) i. organization of the flight motor. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, **312**(1158), 527–551.
- Wang, Z. (1990a). 2-MOSFET transresistor with extremely low distortion for output reaching supply voltages. *Electronic Letters*, **26**(13), 951–952.
- Wang, Z. (1990b). Current-controlled linear MOS earthed and floating resistors and their application. *IEEE Proceedings on Circuits, Devices and Systems*, **137**(6), 479–481.
- Wang, Z. and Guggenbuhl, W. (1990). A voltage-controllable linear MOS transconductor using bias offset technique. *IEEE Journal of Solid-State Circuits*, **25**(1), 315–317.
- Wanlass, F. M. (1967). Low stand-by power complementary field effect circuitry. *U.S. Patent No. 3,356,858*, .
- Wanlass, F. M. and Sah, C. T. (1963). Nanowatt logic using field-effect metal-oxide semiconductor triodes. *International Solid State Circuits Conference Digest of Technical Papers*, pp. 32–33.

- Warren, D. H. and Strelow, E. R. (1985). *Electronic Spatial Sensing for the Blind: Contributions from Perception*. Springer.
- Wee, K. H. and Sarpeshkar, R. (2008). An electronically tunable linear or nonlinear MOS resistor. *IEEE Transactions on Circuits and Systems I*, **55**(9), 2573–2583.
- Wehner, R. (1981). Spatial vision in insects. *in* H. Autrum, ed., *Handbook of Sensory Physiology*. Springer Verlag. Berlin/Heidelberg. pp. 287–616.
- Wikipedia (2011). Optical flow. Available online at http://en.wikipedia.org/wiki/Optical_flow; visited on November 27th 2011.
- Wikipedia (2012a). Johnson-Nyquist noise. Available online at http://en.wikipedia.org/wiki/Johnson%E2%80%93Nyquist_noise; visited on March 14th 2012.
- Wikipedia (2012b). Shot noise. Available online at http://en.wikipedia.org/wiki/Shot_noise; visited on March 14th 2012.
- Yang, Z., Gruev, V. and Van der Spiegel, J. (2006). A CMOS linear voltage/current dual-mode imager. *IEEE International Symposium on Circuits and Systems*, pp. 3574–3577.
- Zingg, S., Scaramuzza, D., Weiss, S. and Siegwart, R. (2010). Mav navigation through indoor corridors using optical flow. *in* *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE. pp. 3361–3368.
- Zufferey, J.-C., Beyeler, A. and Floreano, D. (2010). Autonomous flight at low altitude using light sensors and little computational power. *International Journal of Micro Air Vehicles*, **2**(2), 107–117.
- Zufferey, J. C., Klaptoch, A., Beyeler, A., Nicoud, J. D. and Floreano, D. (2007). A 10-gram vision-based flying robot. *Advanced Robotics*, **21**(14), 1671–1684.
- Zufferey, J. and Floreano, D. (2005). Toward 30-gram autonomous indoor aircraft: Vision-based obstacle avoidance and altitude control. *in* *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE. pp. 2594–2599.
- Zufferey, J. and Floreano, D. (2006). Fly-inspired visual steering of an ultralight indoor aircraft. *Robotics, IEEE Transactions on*, **22**(1), 137–146.

Zufferey, J., Klapotocz, A., Beyeler, A., Nicoud, J. and Floreano, D. (2006). A 10-gram microflyer for vision-based indoor navigation. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3267–3272.

Curriculum vitae

PERSONAL INFORMATION

Name	Rico Möckel
Date of birth	02.02.1981
Nationality	German

AWARDS

2010	My student D. Jäckel won the paper award "Silver Leaf Certificate" for our joined paper submitted to IEEE PRIME conference
2007	Presentation First Place and Gold Medal at international Competition on Genetically Engineered Machines (iGEM) 2007
2005	Industrial Robot Highly Recommended Award
2002	Best engineer student award from The Association of German Engineers (VDI Mecklenburg-Vorpommern)

EDUCATION

Sep. 2005 - present	Ph.D. student at the Institute of Neuroinformatics, University of Zurich and ETH Zurich
Dec. 2004 - Jun. 2005	Diploma thesis at the Institute of Neuroinformatics, University of Zurich and ETH Zurich
Oct. 2000 - Jun. 2005	Studies of Electrical Engineering at University of Rostock, Germany
1999	Abitur (university entrance level)
1987 - 1999	Primary school and Gymnasium (12 year secondary school) in Schwerin, Germany

PROFESSIONAL EXPERIENCE

Apr. 2009 - present	Research Assistant at EPFL Biorobotics Laboratory, Switzerland
Aug. 2008 - Jan. 2009	Teaching Assistant, "Neuromorphic Engineering I" at University of Zurich and ETH Zurich
Feb. 2008 - Jul. 2009	Teaching Assistant, "Neuromorphic Engineering II" at University of Zurich and ETH Zurich
Apr. - Sep. 2004	Internship at EPFL Biologically Inspired Robotics Group, Switzerland
Jan. 2002 - Nov. 2004	Student assistant / student research projects at Department of Electrical Engineering, University of Rostock, Germany
Feb. - Mar. 2001	Assistant in sales and marketing department at Siemens Rostock, Germany
Aug. - Sep. 2000	Technical assistant at Pirelli Cable and Systems in Schwerin, Germany
Nov. 1999 - Aug. 2000	Basic military service and staff officers assistant in Schleswig, Germany
Sep. - Oct. 1999	Technical assistant at KGW Maschinenbau GmbH in Schwerin, Germany

SCIENTIFIC PUBLICATIONS

Journal publications

A. Spröwitz, S. Pouya, S. Bonardi, J. van den Kieboom and R. Möckel et al. "Room-bots: Reconfigurable Robots for Adaptive Furniture", IEEE Computational Intelligence Magazine, special issue on "Evolutionary and developmental approaches to robotics", 2010.

A. Sproewitz, R. Moeckel, J. Maye, and A.J. Ijspeert. "Learning to move in modular robots using central pattern generators and online optimization". International Journal on Robotics Research, 27(3-4):423-443, 2008.

R. Moeckel, C. Jaquier, K. Drapel, E. Dittrich, A. Upegui, and A.J. Ijspeert. "Exploring adaptive locomotion with YaMoR, a novel autonomous modular robot with Bluetooth interface". Industrial Robot, 33(4):285-290, 2006.

Book chapter

R. Moeckel, and S.-C. Liu. "Motion detection chips for robotic platforms". In "Flying

Insects and Robots”, Floreano, D., Zufferey, J-C., Srinivasan, M. and Ellington, C (Eds.), Springer Berlin Heidelberg, 101-114, 2009.

Conference publications

S. Bonardi, J. Blatter, J. Fink, R. Moeckel, P. Jermann, P. Dillenbourg, and A. J. Ijspeert. ”Design and evaluation of a graphical iPad application for arranging adaptive furniture”. In 21st IEEE International Symposium on Robot and Human Interactive Communication, 2012, accepted for publication.

S. Bonardi, R. Moeckel, A. Sproewitz, M. Vespignani, and A. J. Ijspeert. ”Locomotion through reconfiguration based on motor primitives for roombots self-reconfigurable modular robots”. In Proceedings of the 7th German Conference on Robotics, 2012.

S. Pouya, M. Khodabakhsh, R. Moeckel, A. Ijspeert, ”Role of spine compliance and actuation in the bounding performance of quadruped robots”, Dynamic Walking, 2012.

A. Spröwitz, A. Tuleu, M. Ajallooeian, R. Moeckel, S. Pouya, P. Löpelmann, M. D’Haene, A.J. Ijspeert ”Design and dynamic gait generation for a small compliant quadruped robot”, Dynamic Walking, 2011.

S. Pouya, R. Moeckel, F. Peuker, A. Seyfarth, A.J. Ijspeert ”Stability augmentation of slip-like legged locomotion exploiting hip actuation”, Clawar, 2011.

F. Peuker, S. Grimmer, S. Pouya, R. Moeckel, A.J. Ijspeert, A. Seyfarth, ”Anchoring the SLIP template: The effect of leg mass on running stability”, AMAM 2011.

A. Sproewitz, L. Kuechler, A. Tuleu, M. Ajallooeian, M. D’Haene, R. Moeckel, A.J. Ijspeert, ”Oncilla robot: a light-weight bio-inspired quadruped robot for fast locomotion in rough terrain”, AMAM, 2011.

A. Sproewitz, P. Laprade, S. Bonardi, M. Mayer, R. Moeckel, P-A. Mudry and A.J. Ijspeert, ”Roombots - Towards Decentralized Reconfiguration with Self-Reconfiguring Modular Robotic Metamodules”, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010.

D. Jäckel, R. Moeckel and S-C. Liu, ”Sound Recognition with Spiking Silicon Cochlea and Hidden Markov Models”, IEEE PRIME, 2010. Awarded with Silver Leaf Certificate.

R. Moeckel and S-C. Liu, ”A time-of-travel motion detection chip for steering autonomous robots”, Workshop on Visual guidance systems for small autonomous aerial vehicles at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008), September 2008.

A. Sproewitz, R. Moeckel, J. Maye, M. Asadpour and A.J. Ijspeert. ”Locomotion in Modular Robots based on Central Pattern Generators”, In AMAM 2008 abstracts, pages

86-87, June 2008.

R. Moeckel, R. Jaeggi, and S-C. Liu, "Steering with an aVLSI Motion Detection Chip", IEEE International Symposium on Circuits and Systems (ISCAS 2008), pages 1036-1039, May 2008.

S-C. Liu and R Moeckel, "Temporally learning floating-gate VLSI synapses", IEEE International Symposium on Circuits and Systems (ISCAS 2008), pages 2154-2157, May 2008.

A. Sproewitz, R. Moeckel, J. Maye, M. Asadpour, and A.J. Ijspeert. "Adaptive locomotion control in modular robotics". In Workshop on Self-Reconfigurable Robots/Systems and Applications (IROS 2007), pages 81-84, November 2007.

R. Moeckel, A. Sproewitz, J. Maye, and A. Ijspeert. "An easy to use bluetooth scatternet protocol for fast data exchange in wireless sensor networks and autonomous robots", In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2007), pages 2801-2806, November 2007.

R. Moeckel and S-C. Liu, "Motion detection circuits for a time-to-travel algorithm", IEEE International Symposium on Circuits and Systems (ISCAS 2007) 3079-3082, May, 2007

R. Moeckel, C. Jaquier, K. Drapel, E. Dittrich, A. Upegui, A. Ijspeert, "YaMoR and Blue-move - an autonomous modular robot with Bluetooth interface for exploring adaptive locomotion." in 8th International Conference on Climbing and Walking Robots (CLAWAR 2005), 685-692, September 2005. Awarded with Industrial Robot Highly Recommended Award

A. Upegui, R. Moeckel, E. Dittrich, A. Ijspeert, and E. Sanchez, "An FPGA dynamically reconfigurable framework for modular robotics." In U. Brinkschulte, editor, Workshop Proceedings of the 18th International Conference on Architecture of Computing Systems 2005 (ARCS 2005). VDE Verlag, Berlin, 2005.

Abstracts

R. Moeckel, and S-C. Liu. "Steering with an aVLSI motion detection chip", Symposium of the Neuroscience Center Zurich (ZNZ), September 2008.

S-C. Liu, and R. Moeckel. "Temporally learning floating-gate VLSI synapses", Symposium of the Neuroscience Center Zurich (ZNZ), September 2008.

R. Moeckel, and S-C. Liu. "Obstacle avoidance with an aVLSI motion detection chip", Symposium of the Neuroscience Center Zurich (ZNZ), September 2007.

R. Moeckel, and S-C. Liu. "aVLSI Motion Detection Circuits", Symposium on Flying Insects and Robots, Monte Verita, Ascona, Switzerland, August 2007.

R. Moeckel, and S-C. Liu. "aVLSI Motion Chip for Microflyers", Symposium of the Neuroscience Center Zurich (ZNZ), October 2006.