Diss. ETH No. 22500

Event-Based Machine Vision

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH
(Dr.sc.ETH Zurich)

presented by

CHRISTIAN PETER BRÄNDLI

MSc. Interdisciplinary Sciences ETH
born on 2. Feb. 1985
citizien of Rorbas, Zurich, Switzerland

accepted on the recommendation of

Prof. Dr. Tobias Delbruck
Prof. Dr. Fumiya Iida
Dr. Shih-Chii Liu

2015

Abstract

Movie cameras were originally developed to create movies for human interpreters but with the dawn of digital cameras their output is now also used for scene analysis using computers. This scene analysis is not efficient because capturing a dynamic scene with a stroboscopic sequence of image frames leads to lot of redundant information. This thesis describes a machine vision approach which relies on sensors that process information based on efficient smart pixels that create events instead of images. An overview over existing event-based vision sensors, algorithms and applications is given. The development of a novel vision sensor optimized for event-based machine vision, the dynamic and active pixel vision sensor (DAVIS) is documented in detail. Furthermore the following four algorithms are described in detail: an event-based vision algorithm for depth estimation using structured light, a novel event-based keypoint detection and description algorithm, a novel event-based, low-level line segment feature and a novel event-based video decompression algorithm for the DAVIS data.

Keywords: Event-Based, Machine Vision, Neuromorphic Engineering, Bio-Inspired, Vision, Dynamic Vision Sensor (DVS), Dynamic and Active Pixel Vision Sensor (DAVIS), Silicon Eye, Smart Pixel, High-Speed, Low-Power, High Dynamic Range

Zusammenfassung

Kameras wurden ursprünglich erschaffen um Filme für menschliche Betrachter zu kreiren, doch seit der Enstehung von digitalen Kameras werden diese Daten vermehrt gebraucht um visuelle Szenen mit Computern zu analysieren. Diese visuellen Analysen sind nicht effizient weil beim Aufnehmen einer dynamischen Szene als stroboskopische Sequenz von Bildern viel redundante Information entsteht. Diese Doktorarbeit beschreibt einen Ansatz für maschinelles Sehen der auf Sensoren beruht welche Informationen mit intelligenten Pixeln verarbeiten und sogenannte Events an Stelle von Bildern kreieren. Sie gibt einen Überblick über bestehende Event-basierte visuelle Sensoren, Algorithmen und Anwendungen. Zudem ist die Entwicklung des sogenannten visuellen Sensoren mit dynamisch und aktiven Pixeln (DAVIS) dokumentiert; dieser neuartige Sensor ist optimiert für Event-basiertes maschinelles Sehen. Des weiteren sind die folgenden vier Algorithmen im Detail dokumentiert: Ein neuer Event-basierter visueller Algorithmus für die Tiefenschätzung mittels strukturiertem Licht, ein neuartiger Event-basierter Algorithmus zur Entdeckung und Beschreibung von charakteristischen Bildpunkten, ein neuartiges Event-basiertes, einfaches Liniensegment-Merkmal und ein neuartiger Event-basierter Algorithmus zur Video-Dekomprimierung der Daten des DAVIS Sensors.

Contents

Acknowledgements

This thesis is the result of a long journey and I want to thank everybody who supported and helped me on the way:
My parents Urs-Beat and Nickey for giving me the freedom to become whatever I want and supporting me in all of my decisions. For their emotional and financial help which is the basis for all I did.
My brothers Martin and Ruben for their loyalty and for having two guys on my side no matter what happens. Martin for being the best intellectual sparring partner I ever had and Ruben for being the best accomplice I could imagine and for the layout of this thesis.
My girlfriend Denise for all of her love and for helping me up after any sort of fallbacks; meeting you is the best thing that ever happened to me.
My supervisor Tobi for being the best supervisor I could have wished for. For the liberty and options I got, for the advise and help, and for the great atmosphere he spread.
The Sensors group for fruitful discussions, ideas and support in my work. Particularly Rapha for providing a substantial part of the work described in this thesis.
All the students I supervised: Matthias Hofstetter, Thomas Mantel, Markus Turnherr, Jonas Strubel, Varad Gunjal, Luca Longinotti, Susanne Keller, Jon A. Lund for their contribution to my research and this thesis.
Marc for initiating my latest adventure in form of our start-up Insightness.
The Institute of Neuroinformatics, the University of Zurich and the ETH Zurich for providing the institutional structures that enabled this work.
My defense committee for reviewing and improving this thesis.
Dennis lunch group, the Castrol Stadium Foosball League and the neuromorphic engineering community for interesting discussions and a good time.
The Rugby Union Zurich for providing me compensation for the otherwise overly intellectual research and for great experiences.
The Zürcher Südkurve, the FC Zürich and the Boys for magic moments.
All The Support for the fun times.

This work has been supported by the Swiss National Funds (SNF) through the National Competence Center in (NCCR) Robotics and the European Union through the seventh Framework Program (FP7) project SeeBetter.

1 Introduction

It was 1872 when Leland Stanford, a Californian tycoon, the ex-governor and university founder decided to investigate whether a horse has all four legs off the ground while trotting. This was a popularly debated question at this time because the human eye cannot perceive such fast motions. Stanford had taken the position that there are so-called unsupported transits (all four legs in the air) in the trot and to prove this, he hired Eadweard Muybridge, an English Photographer (Mitchell 2001).
After several failures and improvements to the shutter used to capture the images, Muybridge was capable of proving the unsupported transit theory, by capturing an image of a horse with all four legs in the air. This success led the two to continue their investigations and start studying the gallop. Muybridge set up 12 cameras along a race track which were triggered by wires stretching across the race track. The now famous series of pictures shows a galloping horse (Fig.1.1) and can be considered to be the first motion picture ever recorded. A year after taking these pictures, Muybridge also developed an apparatus to display these images on a screen: By modifying the zoetrope, a kid's toy at that time, he created a projector called zoopraxiscope that allowed watching the images as an animated movie.

The First Movies

The main drawback of the method and reason why the "horse in motion" is often not considered to be a real movie, is the fact that it was captured with multiple cameras and the pictures had to be copied with hand drawings so that they could be displayed with the zoopraxiscope. And even though the French celebrate the brothers Lumière as inventors of the cinema and the Americans claim the same for Thomas Edison, it was Louis Le Prince who shot the first movie using a camera with a single lens. In 1887 Le Prince built the first single-lens camera which was used to shoot the "Round Garden Scene", a 2.11s long movie with 52 frames showing people walking around in a garden. In follow-up of the invention of the movie camera, the cinemas that started to spread among the cities showed silent films, usually accompanied by an orchestra. Even though the first projections of sound films already took place in 1900, sound films were not immediately a commercial success because it was hard to synchronize and amplify the recorded sound. Only with the invention of the sound-on-film technique, which captured the sound on the same film, sound film and the so called "talkies" became the industry standard at the end of the roaring twenties.

Color Movies

The first color movies which were shown 1895 in Thomas Edisons so-called Kinetoscope were monochrome pictures colored by hand; later hand-coloring was replaced by automated coloring systems such as the "Pathe Color" in 1905. The first color motion picture process was the RG two color additive Kinemacolor process that was inspired by the pioneering work of Edward Raymond Turner. The main drawbacks of this system were the small gammut and the fact that the red and green channel were not exposed at the same time which led to motions artifacts. The Technicolor system which used a beam splitter (prism) allowed exposing the two color channel at the same time and with the development of the subtractive color system and a three color system, they became the industry standard. Even though the color technologies were improved and refined, the actual change to color movies was only launched with the spread of the television (TV) which competed with cinema.

Television

In the year 1884 Paul Gottlieb Nipkow patented a spinning-disk image rasterizer that scanned the light intensities in a scene using a rotating disk with a hole pattern. John Logie Baird then used this principle in combination with halftone still image transmission technology developed by

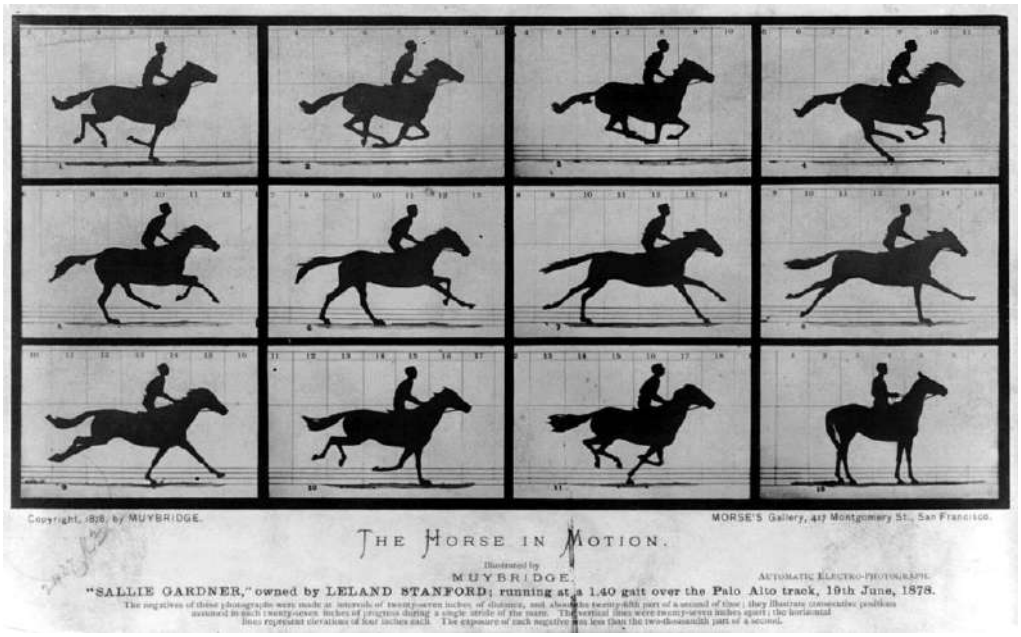


Figure 1.1: Eadweard Muybridge's 1872 horse in motion taken with 12 wire triggered cameras set up along a race track (from (*Wikipedia*)).

Arthur Korn to build the first electromechanical television camera in 1924. Inventions such as the image dissector developed by Philo Farnsworth or the iconoscope allowed capturing the TV signal fully electrically and the cathode ray tube (CRT) allowed to display these signals also fully electronically. Even though Baird demonstrated the first color transmission in 1928 (electromechanical), the first color TV program was only shown in 1953 because the second world war and discussions on standards hindered a faster advancement of the technology. When Ampex released their Ampex VRX-1000 in 1956 video recorder, it was for the first time possible to record TV signals. This recording technology together with video camera tube (vidicon) then allowed NASA/JPL to record the first digital images in their space probes by storing video still images and transmitting them to earth. But since the images were acquired with a scanning technique and not captured with an array of photosensitive elements this technology is often not considered a digital camera.

Digital Cameras

Along with the recording possibilities for video signals, the other enabling technology for digital photography was the charge coupled device technology (CCD). The CCD was in 1969 originally invented as memory technology by Willard Boyle and George Smith (Nobel prize 2009). Michael Tompsett later realized that it can be used to capture images. In 1975 Steve Sasson then combined the CCD chip with a cassette tape recorder to build the first analog electronic camera which was in principle a video camcorder that stored single images. The biggest drawbacks of CCD imagers are: they can only be produced in special, dedicated chip production processes and can therefore not be directly integrated with other circuits. This disadvantage of digital imagers was overcome with the inventions of the active pixel sensor (APS, (Fossum 1993)), in-pixel charge transfer and correlated double sampling which allowed capturing images of acceptable quality using the same chip production processes that are used for digital logic chips. With these inven-

tions, the so called complementary metal-oxide-semiconductor (CMOS) production process can be used to produce chips that contain image acquisition as well as image processing for a small price. With the abundance of more and more digital images and computers with increasing computational power, the field of computer vision gained more and more attention and many applications for computer vision emerged.

Digital Movies

Even though the technology of capturing and analyzing dynamic scenes has changed dramatically since the days of Muybridge, the underlying principle remains the same. Motions and changing scenes are recorded as a stroboscopic series of still images also known as frames. While this way of sampling a scene uniformly in space and time allows employing uniform and easily developed processing routines, it is inefficient. This does not matter as long as time and energy are not key to an application, such as in the field of academic computer vision where algorithms can run several days to analyze a scene and burn megajoules of power. But in systems that interact with the real world, latency becomes an issue and if the systems run only on battery power, power consumption also becomes a critical aspect. For real-time machine vision on mobile systems it is therefore crucial to optimize the processing efficiency and as nature undoubtedly came up with the most efficient visual processing principles, it makes sense to take them as inspiration for artificial vision systems.

A Smarter Approach

The visual processing stream of animals, including humans, does not use the concept of frames. This is for a good reason: instead of sampling all pixels at the same time, the photosensitive cells in the eyes and the underlying processing circuits in the retina decide themselves when there is information to be processed. So instead of reading the same constant value over and over again, it is mostly the relevant changes that are communicated and many cell responses adapt to constant inputs with a decreased activity (Smirnakis et al. 1997). This concept can be understood as on-sensor data compression which decreases the data before it is communicated in the form of spike events to the processing structures (i.e. visual cortex). The goal of my thesis was to investigate how such a neuro-inspired, event-based approach can improve machine vision and to develop event-based machine vision systems.

Thesis Organization

Before I started with my thesis, a lot of work that could be considered event-based machine vision (EBMV) had been published. Most of this work was never termed event-based machine vision but from what I learned during my PhD studies, it makes sense to join the research under a common umbrella. This thesis offered the opportunity to do so and to come up with a description, a framework and a review of the field. I do not claim to have invented the field, but I hope that in the following I can give some arguments on why it makes sense to define it as a field, why it should be named Event-Based Machine Vision and what it is about. Another reason for this unconventional thesis format is the fact that I worked on variety of different topics and methods; this format allows me to cover all of this work as well as linking it to existing research. This is because during my studies I did not focus on one single research question which would allow as a single story but I followed a more explorative approach. I investigated multiple possibilities in this field and the branched structure of this thesis allows to contextualize the findings. The aspects of the field to which I have contributed myself are documented in the "Study" sections of the according chapters. Most of this work was realized in collaborations and to be specific about my own work, I added a "Contributions to the Field" section to the conclusion which gives a detailed overview over my personal contributions.

Thesis Overview

The thesis is structured in the following way: I start off with a description of *Event-Based Computation and Technology* which is the basis for event-based machine vision. I motivate the use of event-based computation and give an overview

over the field of neuromorphic engineering. *Event-Based Vision Sensors* is a chapter on existing sensors as well as on the development of the "dynamic and active pixel vision sensor (DAVIS)" in *Study DAVIS*. In *Event-Based Machine Vision Algorithms*, I give an overview over existing algorithms in the field and document the algorithms developed during my thesis: *Study of Event-Based Structured Lighting*, *Study of DAVIS Decompression*, *Study of Event-Based Line Segment Detector (ELiSeD)*, *Study of Event-Based Keypoints*. The application of event-based algorithms is discussed in *Event-Based Machine Vision Applications* and the thesis finishes off with *Conclusion and Outlook*.

2 Event-Based Computation and Technology

Since a kid, I have been motivated by questioning and reasoning; I wanted to understand everything. This journey of doubt lead me deep into the fields of philosophy and from there through science to engineering. The insights I gathered on my way would be out of scope for this thesis but in this chapter I try to contextualize the presented work in a bigger picture because I am convinced that bad scientists do not question their reasoning and bad philosophers do not reason their questioning.

The way personal computers (PCs) operate is principally different from the way our brain performs computations. Even though many aspects of neural computation are not yet understood, insights into the basic principles indicate four main differences: nervous systems are analog, asynchronous, parallel and self-organized circuits while most computers are digital, clocked, serial circuits which are programmed to execute a predefined code. In the late 80's Carver Mead, a Caltech professor and Silicon Valley icon, together with his colleagues realized that it should be possible to use the same technology which is used in conventional computer chips (also known as integrated circuits, IC) to produce circuits with characteristics similar to biological neural networks. Driven by this idea, Mead set up a research group to investigate and characterize such bio-inspired artificial nervous circuits. From these first efforts field of neuromorphic engineering (Mead 1989) emerged and eventually resulted in a novel type of event-based sensors. These sensors are the basis for all of the work presented in this thesis.

To contextualize the field of event-based machine vision (EBMV), it is important to understand the nature and history of computation. The field is about symbolic computation but it uses a principle inspired by neural computation. To understand these two fields, this chapter gives an introduction to the field of symbolic computation and neural computation, highlights the main differences between them and introduces the field of neuromorphic engineering. This chapter also highlights the advantages of event-based computation and concludes with a definition of event-based machine vision.

2.1 Symbolic Computation

To understand the differences between symbolic computation and the way brains or brain-inspired circuits compute, it is important to understand what "computation" is as well as the principles by which modern computers work. In the following these principles are explained and their characteristics and advantages are highlighted. Many terms in the following section are only described or outlined but not strictly defined because most definitions rely on undefined terms or contrived distinctions which unnecessarily hamper the understanding[1]. Symbolic computation refers to all computation that is based on manipulating abstract symbols (such as numbers, words or variables). It is different from neural computation which is performed by nervous systems that can solve problems without the requirement for abstract symbols.

2.1.1 The Nature of Symbolic Computation

The success of the human species is based on the fact, that we did not evolve to fit into a certain environmental niche but we are shaping the environment so that it fits our needs. We can manipulate our environment because we learned that it is follows certain rules which we can describe

[1] Even though mathematicians and lawyers might not like it but any definition is either incomplete or does not relate to something in the world we perceive. Any use of language or other symbolic communication relies on the assumption that low-level abstractions such as color or shape are shared among individuals. So either a set of the definitions contains these non-definable abstraction steps or this abstraction is avoided by only using definitions within a symbol space that does not relate to the real world.

and communicate with symbols. These conceptions and models allow us to modify the world according to our needs and they have evolved over thousands of years and became more and more powerful in describing the world. Most of these models, rules and theories can be categorized into the following disciplines:

- **Logic:** Logic rules guarantee that a set of statements[2] does not contradict itself and that a set of true statements does not produce a wrong statement; they ensure consistency and validity[3] of theories. Logic is the basis of all models because if a model would produce two contradicting or wrong outcomes, it is worthless in describing the world and thus useless in manipulating it.

- **Mathematics:** Mathematical rules guarantee the consistency and validity of comparative statements. While most mathematics concerns the comparison of quantities expressed in numbers, it also covers comparisons of spatial or temporal relations. Mathematics are a substantial part of many models because the world is in most cases described using quantities and spatial or temporal relations.

- **Sciences:** Science guarantees the consistency of statements, models and theories with phenomena in the real world; it guarantees their truth. While the logical consistency and validity of a statement can be proven, this is not possible for its truth. Proving the truth of a statement means guaranteeing that there is no phenomenon that contradicts it. This would require direct access to all information in the world. Unfortunately we humans do not only have a very limited access to the phenomena in the world but it is also very indirect (i.e. through our senses and via experiments). No statement is therefore absolutely true (under all conditions) but a true statement means that it is not yet falsified by a contradicting observation. For this reason, a model can only be considered scientific if its sentences are consistent with the observed phenomena at all times i.e. when it is testable and repeatable. Another requirement for a scientific theory is of more practical nature: a scientific theory should rely on the smallest number of parameters in its explanation (Occam's razor). Science is therefore the discipline of creating knowledge i.e. true statements and theories about the world.

- **Engineering:** Engineering guarantees the validity, consistency and truth of statements, models and theories that describe manipulations of the world. By applying logic, mathematics and science in manipulations of the world, it can repeatably be manipulated with the intended outcome. Science delivers insights into how the world works and engineering uses these insights to create methods and tools that allow solving a particular problem. Engineering is therefore the discipline of creating solutions i.e. producing powerful tools to modify the world according to a need.

The borders between these disciplines can not be drawn clearly but the given descriptions should serve as guidance to outline them. Of course many great scientists have been great engineers and mathematicians because they needed built their own tools and methods to investigate the world (e.g. Galileo's telescope, Newtons mathematical achievements, Pasteurs pasteurization, Curie's electrometer, ...). And great engineers usually have a good understanding of science and math.

The more cases a scientific or engineering theory covers, the more general and though the more powerful it is but if it is applied to a specific case, symbolic computation is needed. In the following symbolic computation is understood as the process of applying logic, mathematical, scientific or engineering rules to a set of input arguments and finding the according set of outputs. Computation itself is therefore any mapping process within a system from its input to its output, from a question to its answer or from a problem to its solution. This definition of

[2] Including formulas

[3] Even though "true" and "false" are often used in the field of logic, it would be more correct to use "valid" and "invalid" because logic covers validity and not truth

computation is wider than most conventional definitions because it should not only cover symbolic computations but also neural computation.

If sensory perceptions are considered as input and muscular activity as output, our brain is constantly computing our actions. Actually computing a behavior optimized for survival can be considered the sole purpose of our brain. But instead of using Logic and Mathematics to compute, the brain performs an informal way of computing by extracting and optimizing it's own set of rules to describe the world. These rules are learned when the brain is exposed to sensory information and they span and generalize the whole input space of a sensory modality as well as any possible sensory combination. These cortical representations of the world allow higher animals to compute meaningful behavior in highly complex situations.

Through social interactions we can share cortical representations by creating symbols and teaching others about their meaning so they associate the cortical representation with a symbol. Early forms of symbols were gestures and sounds which later evolved into spoken words, icons and paintings, letters, numbers, words and formulas. As humans started to exchange the ways their brains represent the world, symbol combinations evolved into Logic, Math, Science and Engineering. Symbols also allow us guiding the computation and associations in our brain in a formalized way i.e. by executing an algorithm. Algorithms are the basis of symbolic computation: they are instructions that explain how to compute a result for a formalized theory. Often the result for a specific theory can be computed in various ways and multiple algorithms and implementations can exist. This is possible because the physical entities that are subject of a statement can be converted and manipulated in various representations and a cortical representation can be associated with multiple symbols.

The most common symbolic representation of physical entities are numbers and one of the simplest implementations of numbers are fingers. So already when we are kids we learn to represent abstract entities with symbols as well as a very simple algorithm: counting. The first step for counting is abstracting a set of sensory inputs into a set of entities. This step is performed by our brain which maps the sensations into cortical representations. While this step seems trivial for humans, it is a very challenging task when performed by artificial systems because the boundaries that we use for separating objects require a physical model of the object and this hard to extract from visual information only. After the entities are identified, a mental algorithm is executed so that sequentially for each entity the numerical representation is increased i.e. for each uncounted object a finger is raised. This very simple example of a mental algorithm already exhibits all characteristics of an algorithm.

2.1.1.1 Algorithms

There have been several attempts in describing what an algorithm is but because of their simplicity the following refers to the five characteristics given by (Knuth 1998):

- **Finiteness:** "An algorithm must always terminate after a finite number of steps."
- **Definiteness:** "Each step of an algorithm must be precisely defined; the actions to be carried out must be rigorously and unambiguously specified for each case."
- **Input:** "An algorithm has zero or more *inputs*: quantities that are given to it initially before the algorithm begins, or dynamically as the algorithm runs."
- **Output:** "An algorithm has one or more *outputs*: quantities that have a specified relation to the inputs."
- **Effectiveness:** "An algorithm is also generally expected to be *effective*, in the sense that its operations must all be sufficiently basic that they can in principle be done exactly and in a finite length of time by someone using a pencil and paper."

This description as many others (e.g. (Davis 1958; Berlinski 2001; Venn 2007)) does not specify anything about how an algorithm has to be executed or implemented in a computing machine. But for the execution of an algorithm its implementation and the computing machine on which it is executed are critical.

2.1.2 Computing Machines

Because algorithms can be very complex and many intermediate results may have to be computed and stored, they are very prone to mistakes and errors. For this reason people started inventing computing machines that facilitate computation and increase its correctness. The following paragraphs give an overview over the development of computing machines and motivate the idea of brain-like computers.

2.1.2.1 Mechanical Computing Machines

The abacus can be considered the first computing machine: they were digital [4] and built to execute a fixed, serial algorithm. While these computing machines facilitate the execution of an algorithm by storing intermediate results, the algorithm itself has to be executed by a human operator.

The first known integration of an algorithm into a machine was the Antikythera mechanism 205BC (Freeth et al. 2006) which allowed to model stellar compositions using a complex assembly of bolts and gears. The first numeric computing machines were the hand operated computing machines of Blaise Pascal 1642 (Marguin 1994) and Gottfried Wilhelm Leibniz 1694 (Marguin 1994) (inventor of the binary numeral system) which integrated arithmetic rules in the form of mechanical concepts. These machines were very useful but could only execute one distinct algorithm. The first (fully mechanical) general-purpose computer was described by Charles Babbage in 1837 (Weber 2000) but it was actually never built. In the end of the 19th century analog mechanical computing machines which did not require a discretized or numerical input allowed to predict the tide or compute differential equations (Thomson 1881). By using physical entities and physical laws for their computation, they could compute very complex relations but they were highly specialized to a specific task.

[4] Digital in the original meaning of the word: based on digits; not binary or electronic

2.1.2.2 Electronic Computing Elements

Mechanical computing machines have several drawbacks: They require a lot of space, they are prone to mechanical stress and wear, data communication over long distances is complicated and it is hard to interface them with optical inputs or outputs. With electrons as information carriers it is much simpler to shrink the computation elements because they can be produced with less mechanical constraints i.e. no parts have to move. Even though electronic circuits also wear out, this happens on a much longer time scale so they require much less maintenance. Electronic signals can cover long distances in very short times and with wires as signal guides it is easy to route them efficiently around corners or if released into the into the ether as electromagnetic waves they can travel without the requirement for a material substrate. And since part of the electromagnetic wave spectrum is visible by humans, it is also simpler to translate electronic signals into visual stimuli or using visual sensors.

The development of the first electronic devices was driven by the progress in the field of telegraphy which is the long distance transmission of symbolic or textual messages. Visual telegraphy with Smoke, Fires, Flags, Mirrors or Semaphores had the drawback that they depended on the weather conditions and the continuous attention of the receiver. To overcome these problems, electronic telegraphy was developed by Carl Friedrich Gauss and Wilhelm Weber in 1833 and later commercialized by Sir William Fothergill Cooke and Charles Wheatstone in England as well as by Samuel Morse and Alfred Vail in the United States. While the electronic circuits of these devices were relatively simple, they became more complex with the invention of wireless

telegraphy in 1880 by David Edward Hughes who used a spark-gap transmitter to generate electromagnetic pulses. But it is the invention of the triode in 1906 by Robert von Lieben and Lee De Forest independently that started a new era of electronic devices.
The triode is a vacuum tube (also known as valve) that has two terminals (cathode and plate) in between which a current can flow if the cathode is heated up. Between the two terminals there is a third terminal (grid) which can be used to modulate the current of the other two. With this device it is possible to amplify small voltage changes on the grid terminal into stronger current changes. While these devices could already be used to build computing machines, they burned a lot of power, produced a lot of heat, they were builky and unreliable. The inventions of the transistor in 1947 by Brattain, Bardeen and Shockley (Gorton 1998) and the metal oxide semiconductor field-effect transistor (MOSFET) invented by Dawon Kahng and Martin Atalla in 1959 (Kahng 1963) allowed to replace the power hungry vacuum tubes and paved the way for the integrated circuit (IC). In 1952 Geoffrey W.A. Dummer presented his idea of what is commonly known as computer chip: "electronic equipment in a solid block" i.e. a single device that contains all circuits required to perform a certain computation. While the first ICs contained only a a few transistors, the advances in the semiconductor industry allowed to pack more and more circuit elements onto a single chip and the development of structured VLSI design by Carver Mead and Lynn Conway facilitated the design so that today's processor chips have up to several billions of transistors (Casale-Rossi 2014).

2.1.2.3 Electronic Computing Machines

In 1936 Alan Turing set the theoretical basis for the modern computer by introducing the concept of a Turing machine: a tape-operated hypothetical device that can compute any computation which can be described by an algorithm. Together with the concept of binary numbers and the Boolean algebra developed by George Boole, it became possible to build computing machines from simple electronic switches that perform logic operations on binary representations as shown by Claude Shannon and Victor Shestakov (Dasgupta 2014). The first type of switches used in such computers were electromechanical relays which had long switching times, were prone to mechanical problems, burned a lot of power and required a lot of space. The Z3 developed by Konrad Zuse (Zuse et al. 2010) which is the first fully automatic, digital computer used relays and even though Zuse anticipated in his 1936 patent Z3 application that the same storage used for data could also be used to store the machine instructions ("Anmerkungen zum John von Neumann Rechner"), this is now known as von Neumann architecture (Von Neumann et al. 1945). Due to the disadvantages of relays, they got replaced by vacuum tubes such as in the first electronic digital programmable computer Colossus (1943) and from 1955 onwards they were replaced by transistors.

2.1.3 Differences in the Implementation

In the history of computing machines, algorithms have been implemented in various kinds of machines and there are fundamental differences among these implementations. These differences are highlighted in the following because the implementation affects factors such as cost, time, energy or area used to run an algorithm. Understanding these differences is key to obtain the advantages of event-based machine vision.

2.1.3.1 Digital vs. Analog

Even with the same input and output, the representation of internal states of an algorithm can be realized in various ways. One key decision when representing internal states in electronic circuits is the one on whether the analog, physical entities representing them should be discretized or not. The most wide spread discretization is the one of a voltage range into two states which form the basis for a binary system: The ground voltage (0V) represents the digit 0 and the supply voltage represents 1. Such digital systems have

several advantages but also some disadvantages:

+ Most devices used to implement algorithms i.e. to manipulate and store internal states differ in their properties even though they might have been produced with the exact same process. This phenomenon called mismatch leads to the fact that the exact same circuit produces different results for the exact same voltages or currents applied. The advantage of a digital system is the fact that it treats a whole range of voltages the same: all voltages closer to ground are a 0 and the opposite is true for a 1. This reduces the effect of the mismatch among the processing units and thereby allows to reliably reproduce an output across different devices.

– All physical entities are analog which means that if an algorithm should interact with the real world or a human user, it requires an analog-to-digital or digital-to-analog conversion stage at the according input or output. Such a conversion requires energy and time but it would not be necessary if the physical entities would be manipulated without being translated into a digital symbol space.

– Analog states and signals are continuous but when they are discretized, they have to be mapped onto a finite number of states. So each digital representation of an analog state is a tradeoff between precision and resources needed to represent a state. As example: an analog voltage requires a capacitance to be stored but its digital representation requires multiple state-holding elements which use space. Furthermore some computations can be implemented on a few analog element while they require hundreds of digital elements.

2.1.3.2 Synchronous vs. Asynchronous

Not only the representation can be discretized but also time: By applying an external, periodic reference signal, also known as clock, it can be ensured that all states are changing in a synchronized fashion. The motivation to do so is very similar as for discretizing the state representations: the mismatch in the devices. Differences in transistor sizing as well as different temperatures lead to different delays between input and output. If two paths (e.g. circuits) are not synchronized, they are "racing" to be the first to affect the output so the output of the algorithm becomes dependent on the differences in delay along different paths in its implementation; this phenomenon is known as a race condition. So synchronized state changes have following pros and cons:

+ If state changes can only occur during a fixed time window (determined by the clock), and the implementation is designed so that even the longest possible computational delay between two state-holding elements fits into this window, race conditions can be eliminated and output can be reproduced over a pre-defined range of temperatures.

– The clock signal has to reach all processing units coincidentally which limits the size of a design.

– The drivers required to drive the clock a considerable amount of power.

– The maximal clock speed is determined by the slowest processing path which slows down the computation for elements that could run faster and thereby the clock introduces latency.

– In applications with a dynamic input bandwidth the clock frequency is determined by the maximum bandwidth which leads to a unnecessary power consumption for input with non-maximal bandwidth. In other words: The clock may be running even though the input is not changing. This effect can be reduced by using dynamic clock frequency scaling which requires addition circuitry.

2.1.3.3 Serial vs. Parallel

Computations can in principle be executed on a single processing unit (CPU) or distributed among multiple units (e.g. a nervous system,

multiple processors or a parallel processor). If the computation is kept on a single device its instructions have to be run in a serial order while if it's distributed, it can be executed in parallel. Running an algorithm in serial has several obvious advantages (discussed in more detail in (Cassidy et al. 2012)):

+ The way humans think is serial because we can usually only focus on one thing. So if we write the instructions for an algorithm it is much easier to think about them if they are executed serially.

+ The number of computation steps that can be executed in parallel depends heavily on the task. So it is much more efficient to execute them in series on a single but powerful processing unit instead of using multiple parallel processing architectures which are idle most of the time if the task can not be executed in parallel.

− The bandwidth of a serial processing unit can only be scaled by increasing its processing speed whereas the throughput of parallel architectures can also be scaled by the number of units (Rodgers 1985).

− The latency to process information which is acquired or delivered in parallel (e.g. images) grows with the number of information sources (e.g. pixels) whereas parallel architectures can decrease the processing latency of information that can be processed in parallel. Pipelining is a solution to decrease the latency for tasks that cannot be parallelized.

2.2 Neural Computation

While symbolic computation is consciously[5] performed by people or programmed into machines, neural computation is happening in animals as an unconscious process in the nervous system[6]. While symbols and symbolic computation are a product of human society to increase the survival chances of its members by describing, predicting and manipulating the world they live in, neural computation is a result of evolution and allows animals to move, behave and react. Neural computation and the neural networks in which it is performed exist only to compute a behavior (i.e. the orchestration of muscle activities) that allows animals to eat without being eaten and to reproduce. Animals have evolved into creatures that eat plants or other animals but since plants and animals do not fall from the sky (unlike rain and sunlight), animals have to move in order to feed. Plants on the other hand do not have to behave for survival[7]. Controlling motion and thereby increasing the chances of survival and reproduction can be considered the sole job[8] of the nervous system and in the following some aspects of what is so far understood of neural computation will be presented to contrast it to the widely used, symbolic way of computation.

2.2.1 The Nervous System

The outputs of the nervous system are activation patterns for muscles and to create meaningful movements with these muscles, nervous systems require sensory inputs which tell the system something about the environment. Even though behavior can also be generated in many organisms that lack a nervous system (e.g. procaryotes using chemotaxis), multicellular organisms could

[5] subconscious manipulation of symbols is in this thesis considered to be learned and non-algorithmic i.e. neural computation

[6] neural computation in the wider sense does not only include the nervous system but also other behaviorally relevant signaling systems such as hormones

[7] while plants can react to their environment, it is happening on a slower time scale which is in this thesis not considered being "behavior"

[8] other aspects such as social interactions are a consequence of this job

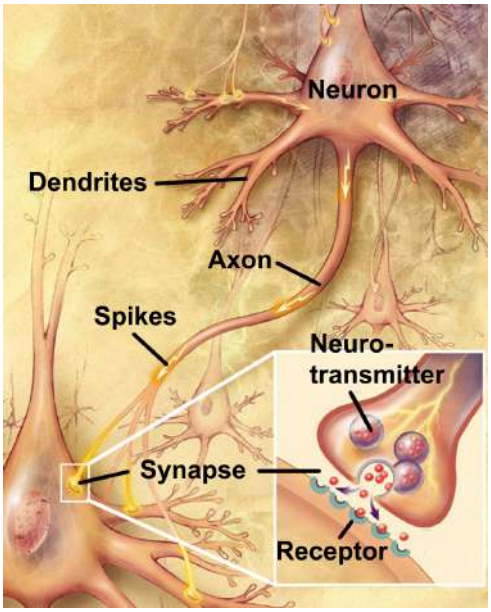


Figure 2.1: The chemical synapse: Spikes travel along the axon to release neurotransmitters which conduct the signal to the receptors of the post-synaptic cell. Adapted from (*Wikipedia*).

not work without a more sophisticated motion control system: the control signals have to travel longer distances than in single celled organisms and the sensory inputs as well as motor outputs are more complex due cell specialization. One challenge that evolution had to solve was signal transduction. It turned out it is more efficient to encode a signal as an ion concentration instead of activating and deactivating proteins as in chemotaxis because ions can move faster and interact over longer distances due to their charge. Interestingly this transition from a mechanical/structural way of signaling to an electronic mechanism can also be observed in the history of symbolic computation because charges can more easy be transported then mechanical signals.

Signal transmission with charges requires that they can be separated and a potential maintained. In the nervous system this charge separation is done across the cell membrane of dedicated cells (neurons) and the signal is encoded by the ion concentration gradients and resulting voltage across the membrane. To connect two or more neurons to each other in a way that allows passing the signal/ion concentration the neurons use synapses. There are two types of synapses: electrical synapses physically link two cells by making their membranes permeable for ions and chemical synapses use communication molecules: neurotransmitters (Fig. 2.1). Chemical synapses translate potential changes into the release of chemical neurotransmitters which then induce changes in the membrane potential of the receiving, post-synaptic cell. Even though the electrical synapse is faster, the chemical synapse has some computationally interesting features: it is directed (only works in one direction), it has an adjustable gain, it has a dynamic response (changing over time) and inhibitory synapses can invert the signal.

To communicate signals over long distances the analog voltages which are prone to noise are converted into digital signals, so-called spikes (also known as action potentials or nerve impulses). Most neurons consist of input branches (dendrites) which can integrate the incoming spikes through synaptic connections of different strengths and as soon as the membrane potential exceeds a certain threshold, a spike is generated and sent along a distinct output branch of the neuron (the axon) through synapses to another neuron. This way sensory signals such as light, sound, pressure or odorants are translated into spikes and then processed by a network of neurons to generate spikes to activate the right muscles for the required action.

For a more detailed and better understanding of the nervous system, the following books can be recommended: (Kandel et al. 2012; Purves et al. 2001).

2.2.2 The Retina

Our eyes and especially the retina served as inspiration for the first event-based vision sensors. For a better understanding of these sensors, the following gives an overview on how retinas work. The retina is the photosensitive neuron tissue of an eye and belongs to the central nervous system. For the retina to work properly, the whole eye

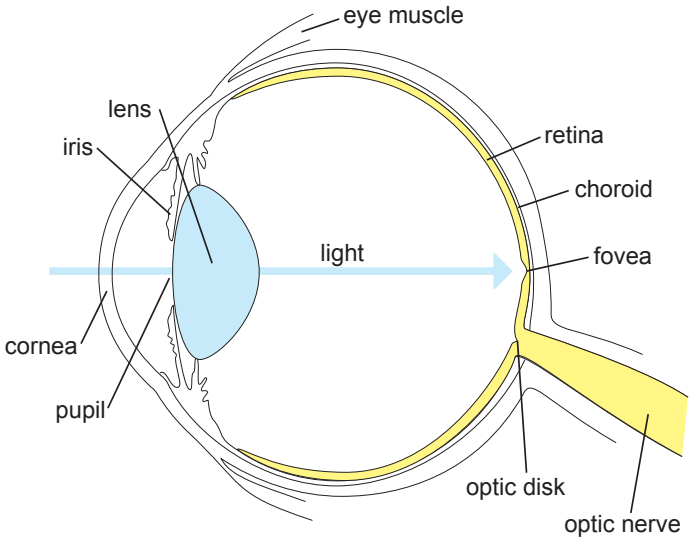


Figure 2.2: Cross-section of an eye: light impinges the eye through the pupil and gets focused and projected onto the retina by the lens. Reprinted from (Lichtsteiner 2006).

(cross-section depicted in Fig. 2.2) is required (Rodieck et al. 1993):

- **Cornea:** The cornea protects the retina and its delicate structures from physical stress and pathogens. In addition it provides most of the refraction of the light so that it is focused onto the retina.

- **Lens:** While the refraction of the cornea cannot be adjusted, the shape and thereby the refraction of the lens can be modulated using the muscles that suspend it. By changing the curvature of the lens, the focal point of the refraction can be set so that the image projected onto the retina is sharp.

- **Eye muscles:** The eye muscles allow to orient the eye along three degrees of freedom: pan, tilt and torsion. This way the fovea, the retinas best resolved area, can be centered at the objects at which the eyes are looking. The eye is actually never resting but continuously scanning a scene for its most interesting features in fast movements called saccades.

- **Iris:** The size of the hole through which light enters the eye, the pupil, can be modulated by a ring-shaped muscle better known as the iris. The outer part of the eye contains pigments which give the eye its color. By changing the diameter of the pupil, the iris adjusts the amount of light falling onto the retina and thereby guarantees that the retina can work as close as possible to the optimal light intensity. The iris therefore has the same function as an aperture in cameras.

- **Choroid:** The choroid tissue contains blood vessels and delivers oxygen and further nourishment to the retina. Fig. 2.3 shows a cross-section of a retina: even though counter-intuitive at first, the photosensitive cells (rod and cones) are not on top of the other cells but on the bottom. This way the light first has to travel through all other cells before it can be detected but this has the big advantage that the photosensitive cells are the closest to the choroid and thereby are nourished the best which improves their response. But a drawback of this arrangement is the

fact that the nerve cells (ganglion cells) that communicate the response of the retina to the brain through the optic nerve travel on top of the retina to the point where they leave the eye: the optic disk (Fig. 2.2). At this point the retina only consists of ganglion cells and does not contain any photosensitive cells so that one is blind at that spot (therefore also referred to as the blind spot).

The retina itself performs a considerable amount of processing (Gollisch et al. 2010):

- **Rods:** The rods are used under low light conditions ("scotopic vision") and produce a monochromatic output with low spatial resolution. Under daylight conditions, the response of the rods is saturated.
- **Cones:** Humans have three types of cones which respond to short (peak at 420nm, blue), medium (peak at 534nm, green) and long wavelengths (peak at 564nm, yellow). The cones work optimally under daylight conditions ("photopic vision") and produce an output that allows to see colors and their high concentration at the fovea allows to the discriminate visual features in the middle of the field of view with a high spatial resolution. In the periphery the rods outnumber the cones about 100 to 1.
- **Horizontal Cells:** Horizontal cells play an important role in the generation of the center-surround receptive fields (Fig. 2.4), color opponency and post-receptoral light adaptation (Thoreson et al. 2012).
- **Bipolar Cells:** There are two types of bipolar cells: the ON bipolar cells are active when illuminated and the OFF bipolar cells are active in the dark. They also show a nonlinear response upon stimulation and they are the basis for the difference between the sustained and the transient response of ganglion cells (Awatramani et al. 2000). The bipolar cells receive their input from the photoreceptor cells (rods and cones) in the outer plexiform layer where also the horizontal cells form their synapses.
- **Amacrine Cells:** For the computation of direction-selective responses and many other nonlinear operations, spatial information has to be integrated which is performed by the amacrine cells that span and integrate many many layers and columns of bipolar cells (Silveira et al. 2011).
- **Retinal Ganglion Cells:** The ganglion cells communicate the information from the retina to the higher brain areas. The receptive fields (stimulus space to which a cell responds) of the retinal ganglion cells show a big variety (Roska et al. 2001) and can be highly specific such as in one distinct ganglion cell type that only responds to approaching objects (Münch et al. 2009). But a common pattern of response is the center-surround response as depicted in Fig. 2.4. The synapses between the bipolar cells, the amacrine cells and the ganglion cells can be found in the inner plexiform layer.

2.2.3 Visual Processing in the Brain

Not only the eyes serve as source of inspiration for event-based machine vision but also the way this information is further processed is often a source of inspiration for vision algorithms. The following gives a short overview over the most important findings on this processing.
As shown in fig. 2.5 the visual information is pre-processed by the retina and leaves the eye through the optic nerve, passes the optic chiasm which splits the two halves of the visual field among the two cortical hemispheres and is relayed at the lateral geniculate nucleus (LGN) which is located in the thalamus (between cortex and midbrain). The LGN is structured into six layers: the inner two layers contain magnocellular (M) cells and the outer four are made up of parvocellular (P) cells. These cells differ in size and exhibit different responses: M cells receive their information from rods, have a transient response, are color-insensitive, exhibit a higher contrast sensitivity at lower spatial frequencies while P cells integrate information from

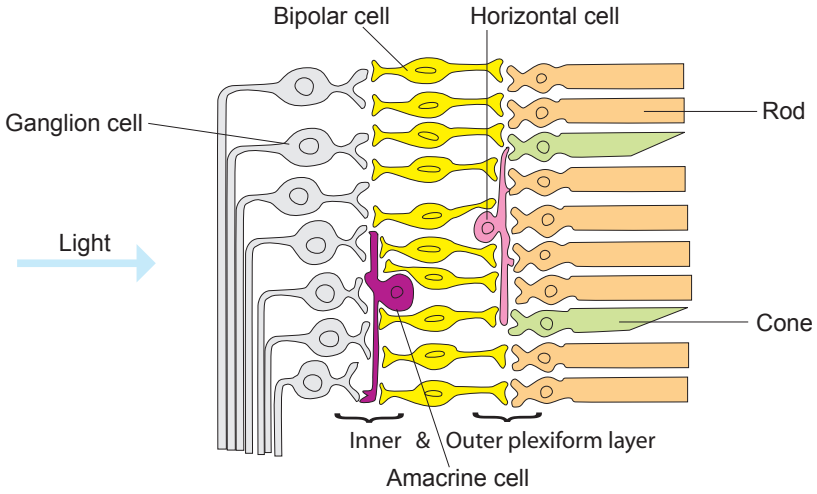


Figure 2.3: The layers of the retina: Light has to travel through the ganglion, amacrin, bipolar and horziontal cells before it is translated into a electrochemical signal in the photosensitive rods and cones. Adapted from (Lichtsteiner 2006).

the fovea, are color-sensitive, have a sustained response and respond to higher spatial frequencies. For these reasons it is thought that already on the level of the LGN, there is a separation of motion and content information. The LGN also receives many feedback connections from the visual cortex and it is thought to decorrelate visual responses (Dan et al. 1996).

From the LGN the visual information is sent to the primary visual cortex (V1) in the occipital lobe of the cortex. V1 follows a retinotopic mapping which means that points that are nearby on the retina, elicit responses at points that are nearby in the cortex. In the first part of their response (first 40ms) many V1 cells exhibit a Gabor-filter-like behavior and they are strongly tuned to simple edge-like stimuli of different orientations. In a later stage the responses are modulated by higher-level feedback connections. It is thought that after this first low-level analysis of orientation and motion of the visual input in V1, the information is further processed in two different streams: the dorsal and the ventral stream (Goodale et al. 1992).

The dorsal stream consists of the dorsal part of V2 and V3, MT, LIP, and VIP areas and terminates in the posterior parietal cortex (PPC). It is considered to be the "vision-for-action" or "where"/"how" pathway because it is relevant for visually guided behaviour, reacts faster and is mostly receiving input from magnocellular cells. The dorsal stream contains an accurate representation of the body, works with egocentric coordinates and has a detailed map of the visual field and is therefore used to interpret spatial relationships. On the other hand it is concerned with analysing, interpreting and guiding movements such as grasping tools (Hebart et al. 2012)

The ventral stream consists of the ventral part of V2 and V3, V4, V5, PIT, CIT, AIT areas. It is considered to be the "vision-for-perception" or "what" pathway because it receives highly detailed information from parvocellular cells, stores long term representations, is relatively slow and relevant for object recognition. It is strongly connected with the limbic system which means that it does not only recognize objects but also emotionally judges their meaning. It is also strongly connected to the dorsal stream and to the temporal lobe which is relevant in memory formation.

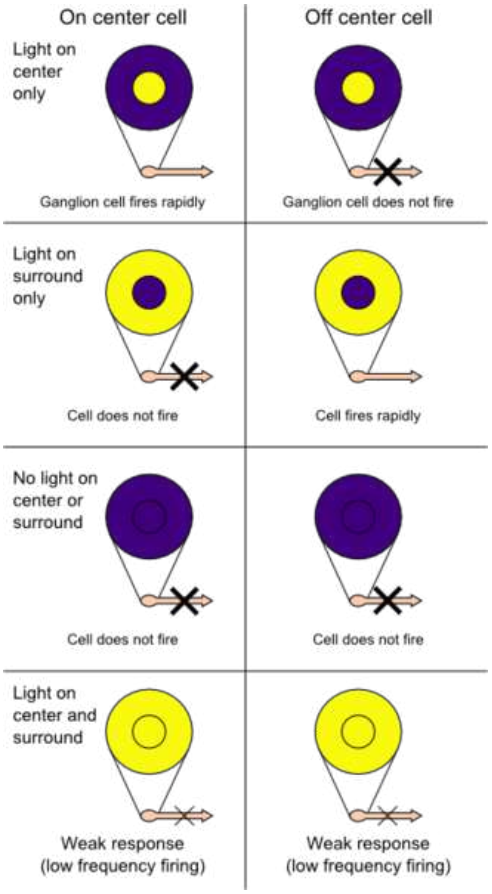


Figure 2.4: The center-surround receptive fields of ganglion cells: depending on their polarity (ON or OFF), ganglion cells respond to specific stimulation patterns that encode spatial contrast . Reprinted from (*Wikipedia*).

2.2.4 Neural Vs. Symbolic Computation

In contrast to neural computation, symbolic computation is not self-contained: it requires abstraction to generate its symbolic input and interpretation to make sense out of its symbolic outputs. In other words, in most cases a computer is worthless unless there is an operator to use it. In the simplest case this abstraction from real world impressions to symbols is an analog-to-digital conversion which maps physical entities onto a predefined scale. Similarly the simplest interpretation of a symbolic output is mapping a symbolic representation to a physical entity (digital-to-analog conversion). Computers are very good in solving problems that can be formulated in symbols and solved with algorithms but they struggle when interacting with the real world. Humans are very good in interacting with the real world and in describing how things in the world relate to each other or how they can be manipulated because it is an evolutionary advantage to do so. On the other hand it is hard for us humans to describe how we abstract these sensory impressions into words and symbols even though we do it very reliably. Investigating how neural computation works is therefore a hard problem: We do not have conscious access to the processes that make up our consciousness. Due to technical and ethical reasons it is also difficult to study these processes in humans and the enormous complexity of nervous systems make it hard to describe them in simple, understandable symbolic rules.

There are algorithms that can abstract symbolic classes from high-dimensional inputs such as images but interestingly some of the best performing of them (artificial neural networks such as deep networks) are very abstract emulations of neural computation structures. And the trick to make these algorithms work is that the actual classification is not hard-coded by an operator but inferred from the presented input data (unsupervised learning) or input-output data pairs (supervised learning) (Bengio 2009). The success of artificial neural networks highlights an important question: what does it mean to understand neural computation? We are now capable of building classifiers with super-human precision but even though we have access to all data in such networks, we cannot say what exactly it does. We can inspect single nodes and describe them mathematically, we can look at network statistics but in the end, the performance arises from the collective interplay of all elements: it is an emergent phenomenon. Even if the function of a single element can be expressed in a single rule, the collective behavior which is responsible for the networks performance can most likely never

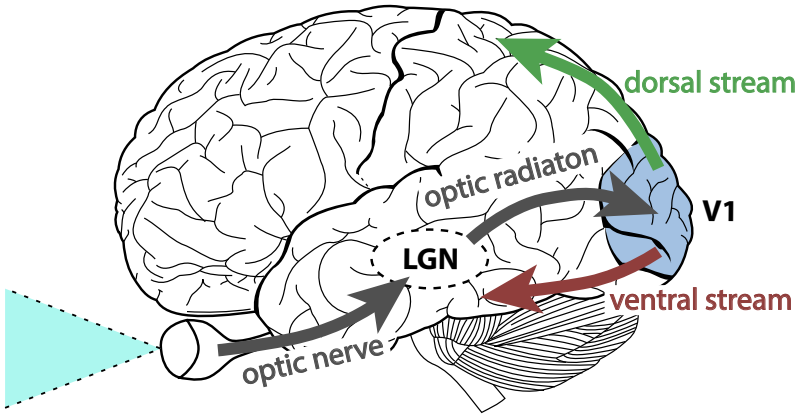


Figure 2.5: Processing Visual Information in the brain: The information from the eye is sent through the optic nerve to the lateral geniculate nucleus (LGN) and from there through the optic radiation to the primary visual cortex (V1). The dorsal stream (green) is specialized for object recognition and the ventral stream (red) specialized to perform object localization. Adapted from (*Wikipedia*).

be described in a single rule[9]. Investigating neural computation means investigating constraints: What learning rules are the most efficient? What network topology performs best for a certain task? What data set allows to extract what type of abstractions? ...

Nature already explored this parameter space through evolution (over thousands of years) and came up with a set of rules that build a network which is capable of learning smart human behavior within a few years. And even though the brain is the result of a long evolutionary development and though it is obviously the most complicated structure mankind knows, there are researchers claiming that it is sufficient to set up a simulation of a population of neurons of a comparable number to come up with a structure similar to the one observed in a brain. The underlying assumption that if the network parameters are chosen right, meaningful computation will emerge is like assuming that it is sufficient just spread out tons of sand and dunes will emerge. But the key ingredient is missing: dynamics; without wind no dunes and without interaction no learning. What shapes the structure of the brain into a smart processing substrate is dynamics: only exposure to the real world allows the underlying structures to identify recurring patterns, classify them and use them to learn the right responses. Unfortunately many large scale simulation and emulations of neural structures do not expose them to the real world so that structures for abstraction and a meaningful interaction can emerge[10].

Independent of the theoretical capabilities of neural computation, it is also highly interesting from an engineering perspective because of its incredible efficiency: The human brain consumes about 20W (Drubach 1999) while it takes Watson, IBMs super computer 80kW (Ledak 2011) to beat humans in Jeopardy and playing Jeopardy is only one of the multiple things our brain is capable of. So if the computational efficiency of neural systems could be replicated, this would be highly interesting. An early effort into this direction was made with creation of the field of neuromorphic engineering.

[9] the same way Windows 8 cannot be compressed to fit on a floppy disk

[10] maybe big amounts of data on the real world would already be sufficient

2.3 Neuromorphic Engineering

Neuromorphic (from neuron = brain cell and morphe = shape, form) Engineering is a discipline of electronic engineering that aims to implement the circuits found in nervous systems in form of silicon circuits. Instead of modeling neural structures as parameters in symbolic computing machines, neuromorphic engineering implements them as physical quantities and thereby does not simulate but emulate the neuron. This means that for instance the membrane potential is not stored as a 32 bit float number in a computer but it is stored as a voltage on a capacitor.

2.3.1 The Origins of Neuromorphic Engineering

The roots of Neuromorphic Engineering (NE) can be traced back to a spring day in 1967 when the Caltech Professor Max Delbrück asked his colleague and expert in transistor physics, Carver Mead on his opinion on a recent claim that transistors used in conventional computers behave like the ion channels of neurons when they are operated in the sub-threshold operation regime (Gilder 2005). This initial contact sparked the interest in Mead to continue investigating the biophysics of membrane channels and their analogy in silicon. In the 1980's he taught the influential "Physics of Computation" course with Richard Feynman and John Hopfield and he started to build first the silicon neuron systems with his group. These circuits included silicon neurons, synapses, retinas, cochleas and further parts of the nervous systems (Mead 1989; Liu et al. 2002; Liu et al. 2014).

2.3.2 Early Silicon Retinas

The first event-based vision sensors were so-called silicon retinas but the first of these neuromorphic vision sensors were actually not event-based.

Silicon retinas are built with the aim to mimic parts of the visual processing that is happening in the retina. Even though (Fukushima et al. 1970) implemented a first silicon retina, research on silicon retinas in the context of neuromorphic engineering began when the biology student Misha Mahowald joined Mead's lab (Mead et al. 1988; Mahowald 1992; Mahowald 1994b). The Mahowald retina was based on a logarithmic bipolar photoreceptor (Mead 1985) that creates an output voltage which is proportional to the logarithm of the light intensity and a resistive network to compute the spatial weighted average of the light intensity in the surround of a pixel (Mead et al. 1988). This spatial contrast sensitive pixel which models the outer plexiform layer of the retina was then simplified and translated into current domain circuits by Boahen and Andreou (Boahen et al. 1992).

Another key topic of early silicon retinas was the development of motion perception (Tanner 1986; Andreou et al. 1991; Moore et al. 1991; Bair et al. 1991; Etienne-Cummings et al. 1992; Horiuchi et al. 1992; Delbruck 1993a; Kramer et al. 1995; Kramer 1996; Kramer et al. 1996; Sarpeshkar et al. 1996; Etienne-Cummings et al. 1997; Etienne-Cummings et al. 1999; Harrison et al. 1999) which are reviewed in (Orchard et al. 2014b).

Based on the findings of a change enhancing photo-receptor (Delbrück et al. 1989; Delbruck 1993b) (later improved in (Delbruck et al. 1994; Delbrück et al. 1996)), a scanned silicon retina that computes temporal derivatives was built (Delbruck et al. 1991) and it can be considered to be a first generation of dynamic vision sensors which are extensively discussed starting on pages 34ff. Other silicon retinas focused on direction selectivity (Benson et al. 1991). An overview over the first silicon retinas can be found in (Koch 1991; Etienne-Cummings et al. 1996; Koch et al. 1996; Sarpeshkar et al. 1996; Spiegel 1996; Indiveri 1999; Stocker 2004).

The first silicon retinas did not include models of the ganglion cells or any sort of spiking output. This is also because a key challenge when implementing the communication of ganglion cells or other spiking neurons in an integrated circuit (IC), is the fact that it is impossible to use dedicated wires for each connection in between

two neurons. Many chips have large numbers of neurons with fanouts in the thousands and the targets might change during operation because of learning. For this reason the spike communication between neurons and between chips has been handled with a dedicated protocol called the address-event representation (Sivilotti 1991).

2.3.3 The Address-Event Representation (AER)

The address-event representation (AER) is an asynchronous event communication protocol widely used in neuromorphic engineering and a key building block to event-based vision sensors. The main idea behind AER is to multiplex the events through an asynchronous bus which only carries the address of the sending cell or pixel (fig. 2.6a). Time-multiplexing for neuromorphic communication is possible because the communication in biological neurons occurs on time scales that are decades smaller than communication in silicon: neurons spike with a few Hertz while most Silicon communication occurs at MHz to GHz.
To accelerate the communication of the asynchronously occurring events, AER is designed as asynchronous data protocol. To avoid incomplete event transmissions in such an asynchronous event protocol, a full handshake protocol is commonly employed (fig. 2.6b):

1. As soon as a sender cell wants to send out an event (e.g. because its membrane potential crossed a threshold), it sets the data valid and requests to write on the address bus by raising the Request signal.

2. When the receiver receives a request, it decodes the address coming with it and transmits it to the target cell. The request can be routed to a single or multiple cells using a router and look up tables. As soon as the receiving cell received the event, it raises the Acknowledge line.

3. The raised Acknowledge line indicates that the receiver got the event and the address must no longer be valid so the sender lowers the Request signal.

4. Upon the arrival of the lowered Request, the receiver lowers the Acknowledge line to indicate it is ready to receive a new event and the sender can raise the Request to start the transmission of a new event.

The first silicon retina which was equipped with event-based AER communication was designed by Mahowald (Mahowald 1992) using circuits from (Sivilotti 1991) and the communication scheme was then further developed by Boahen (Boahen 1996b; Boahen 1997). AER was not only used for the communication of sensor data but also for sensory processing (Lazzaro et al. 1993).

2.3.4 Neuromorphic Engineering Today

Part of the field of neuromorphic engineering is not aiming towards outperforming conventional computing but understanding neural computation by building it. But unfortunately up-to-date, the field couldn't deliver any mayor, novel insights that would have had an impact into neuroscience. On the other hand, the conventional approach to computation is backed by a big industry that drives it to scale fast[11] and it is optimized for mass production, reproducibility and reconfigurability. While it takes at least a year to design and characterize a new neuromorphic chip, software can be implemented and tested in days. So the crucial question to ask is: why implementing neural circuits on a custom-designed chip, when they can be simulated in software with much less effort?
Most answers to this question can be categorized into two classes, the "`learning by doing'" and the "`neuromorphic technology'" type of answer. The "`learning by doing'" answers claims that a direct implementation in hardware will lead to new insights about the physical aspects of computation which a simulation cannot deliver. The

[11] According to Moore's law circuit density-doubling occurs every two years.

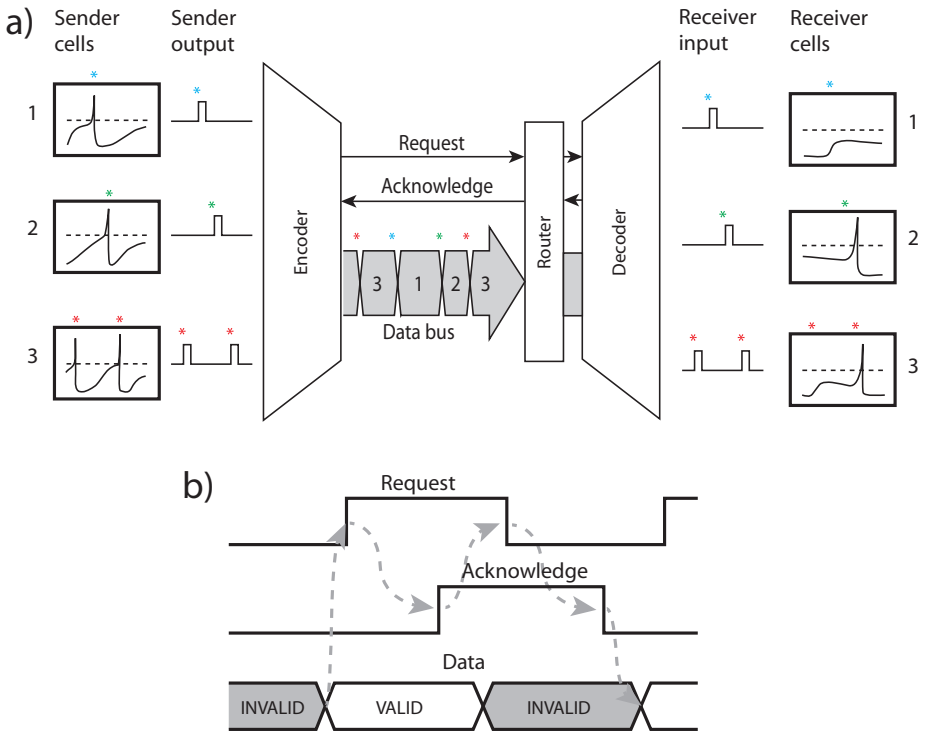


Figure 2.6: Address-event representation (AER): a) An asynchronous bus communicates the address of the sending cell to a receiving cell. b) The 4-phase, active high handshake with bundled data (Liu et al. 2014). Reprinted from (Brandli 2010).

analog and asynchronous nature of the neural computation cannot really be implemented in simulations on digital, clocked computers and crucial aspects of neural computation might be lost if it is discretized in time and states. The main problem with this type of answer is that it opens the door for arbitrariness: Since nobody knows what kind of insight we might get from building neuro-inspired hardware anybody can claim that he is on the right track to get these insights. Novelty becomes the only criterion to judge the quality of the research but just being new does not qualify as science because it does not explain something about the world. Novelty does also not qualify as engineering because it does not solve a problem.

The "`neuromorphic technology"' type of answer on the other hand says that we should get inspiration from the neural computation because it might lead to more efficient computing architectures.

2.3.5 Neuromorphic Technology

Neuromorphic technology can be classified as an engineering discipline (as described above) that produces processing hardware and software which are inspired by the brain, neurons or neural circuits. Neuromorphic technology is designing structures, computational devices or algorithms similar to the ones found in the brain not to learn about the brain but to build systems which can outperform conventional approaches. In contrast to the original Mead approach which aimed to understand the brain by replicating

it, neuromorphic technology is not a scientific discipline that tries to deliver insights into the workings of a real brain but it is a engineering discipline with a clear focus at improving the way we manipulate the world. Neuromorphic technologies usually implement one or multiple brain-inspired computation principles:

- **Asynchronous Processing:** Using clocks is advantageous to simplify the design of circuits and it guarantees the validity of data but it can also lead to unnecessary power consumption. Neurons do not require an external clock for state changes and therefore only burn power when they are involved in computation. This principle can also be adapted to circuits but it requires that data transmission between and within computational elements can be guaranteed without assuming delay (which might change at different temperatures for instance). Alain Martin, Rajit Manohar and colleagues have developed a design flow for such asynchronous quasi-delay-insensitive circuits and they built asynchronous field programmable gate arrays (FPGAs) and microprocessors. (Martin et al. 1997; Teifel et al. 2004)

- **Analog Processing:** Computation in the analog domain without the requirement of digitization is a central aspect of neural computation. If analog computation is done using transistors, it has the advantages that it can be more compact, power-saving and faster compared to digital approaches. A key problem with analog computation in transistors is the mismatch among them. This problem can be tackled using floating gates which permanently store charges to calibrate the circuits. Already in the 1990s the advantages of analog computation have been studied (Sarpeshkar 1997) and today Jennifer Hasler and colleagues are continuing these neuromorphic technologies for instance with field programmable analog arrays (FPAA) (Hasler et al. 2001; Hasler et al. 2013; Brink et al. 2013)

- **Self-Organized Computation:** Computers are pre-programmed and can only execute fixed code determined by the programmer. Nervous systems on the other hand are self-organized, adapt to their inputs and can learn the required computation themselves.

- **Embodiment:** The nervous system and the body it controls have co-evolved so that they are arranged in a "smart" way. Instead of having a general purpose CPU in the middle to control a system with arbitrary degrees of freedom, neural computation is embodied into a highly specific set of sensors and actuators which exploit synergies between body and computation. Most sensory and motor structures are highly optimized to work together to solve highly specific tasks.

- **Event-Based Computation:** Instead of sampling the world with a fixed rate and then updating the internal states and the output at a regular interval, the brain processes information presumably in a bottom-up approach. In the following this approach will be explained in more detail.

Event-Based Machine Vision is a neuromorphic technology which exploits the advantages of event-based visual data representations. It is not a sub-field of neuromorphic engineering because these event-based solutions are not built to understand the brain nor do they have to be implemented in analog or asynchronous bio-inspired processing platforms.

2.3.6 Neuromorphic Technology Companies

A variety of companies emerged from the neuromorphic community and although their products are far away from artificial neurons or brains, they demonstrate the potential of this technology:

- **Synaptics:** Co-founded in 1986 by Carver Mead, Synaptics aimed to develop analog hardware for neural networks. Unfortunately they could not find a suitable market and business model so they used their expertise in analog design to develop a touch pad (they were

asked by Logitech). The touch pad which they designed performed better than other designs and therefore it was highly demanded by the laptop market. The company grew into a 1000 employee company with an annual revenue of almost a billion dollars and about 70% market share of the touch pad market.

- **Foveon:** Also co-founded by Carver Mead, Foveon develops high-end imagers that use all photons to get color images instead of losing most of them to the color filters. First they used a prism to do capture all wavelengths and then they developed the X3 sensor with a photo-diode that allows to infer the color at a pixel from the penetration depth of the photons: blue photons are absorbed at the surface while red photons can penetrate the silicon wafer several microns deep. Even though this approach is technically superior, it did not replace the Bayer color filter which is likely because of a combination of suboptimal marketing, the megapixel race and a low initial performance. Instead of licensing the technology to all big sensor/camera manufacturers, they teamed up with Sigma Corporation, a Japanese lens maker and built full cameras which did not conquer the whole market of digital photography but only a high-end niche. Today Foveon is owned by Sigma.

- **Audience:** In 2000 Lloyd Watts, a former PhD student of Carver Mead who built silicon cochleas, founded Audience with the vision to bring his findings to the mass market. Audience earSmart™technology is inspired by the computation of the cochlea and used to enhance voices and suppress noise in smart phones. Today Audience IP is included in products from Samsung, LG, HTC and Pantech, the company has over 300 employees and a yearly revenues of USD 160M.

- **Achronix:** The company co-founded by Rajid Manohar focuses on the development of asynchronous high-speed FPGAs. They produce their chips at Intel fabs and serve mostly the high-speed communication and network router markets.

- **inilabs:** Founded as a spin-off of the Institute of Neuroinformatics (INI) in Zurich in 2009, inilabs sells Dynamics Vision Sensors to research groups in academia and industry, supports with technical issues, licenses the technology to interested parties and does contract engineering on the technology.

- **Insightness:** Founded in 2014 as spin-off of the INI, based on the technology described in this thesis, Insightness develops high-speed and low-power keypoint-tracking, odometry and simultaneous localization and mapping (SLAM). The targeted markets are mobile robotics and augmented reality on smart glasses.

In addition to neuromorphic companies there are also some specific products that have been influenced by neuromorphic engineering:

- **Logitech Trackball "TrackMan®" Marble:** The TrackMan Marble uses the patented Marble Sensing Technology developed at the Centre Suisse d'Electronique et de Microtechnique (CSEM) which uses a neuromorphic motion detector designed by André van Schaik and colleagues (Arreguit et al. 1996). This system has the advantage that the only moving part is the ball and it therefore does not suffer from dust or dirt problems.

- **WowWee Toys "Robosapien":** Mark Tilden who attended some of the Telluride neuromorphic engineering workshops built a series of biomorphic robots for WowWee Toys like Robosapien, Roboraptor, Robopet, Roboreptile, Roboquad, Roboboa, Femisapien or Joebot. These robots perform complex computations with very simple means using principles of embodiment.

2.4 Event-Based Computation

Event-based computation differs from sample-based computation in multiple aspects and this section will give a short overview over its characteristics and advantages to motivate the next chapters. To simplify the description of event-based systems, the following subsection introduces a new event notation which can be used across all event-based systems.

2.4.1 Event Notation

In biological systems, an event/spike carries only the information on which channel (axon) it arrives and depending on this, the synapses then weighs its relevance by the synaptic strength i.e. a variable amount of current is injected in the post-synamptic cell. In many time multiplexed systems using AER, the axon identity corresponds to the sender address. The other piece of information contained in an event is the time of its creation which is timestamped for systems that do not operate in real-time (e.g. by buffering events into packets for communication efficiency). In the following a way of expressing such events as address and timestamp tuples is proposed.

2.4.1.1 Events

A simple possibility to express events mathematically is to express events Ev as function of address and timestamp:

$$Ev(Ad,t) = \begin{cases} 0, \text{no event} \\ 1, \text{event} \end{cases} \tag{2.1}$$

One notation for the events of the dynamic vision sensor builds on this concept and delivers the polarity (ON or OFF) as function of address and timestamp (Benosman et al. 2011):

$$Ev(x,y,t) = \begin{cases} -1 & \text{if } OFF \text{ event} \\ 1 & \text{if } ON \text{ event} \end{cases} \tag{2.2}$$

This notation is useful for some definitions and allows to easily sum events but unfortunately it has a few shortcomings: It is lacking an index which would allow to order the events according to the sequence in which they were generated and communicated. The notation is inconsistent because it returns one part of the address (polarity) as a function of the another part (x and y address) which seems arbitrary. The notation can further only be applied in systems with discrete timestamps which means that t is not a precise moment in time with infinitesimal precision which would be hard to express in a number but rather a time span e.g. 1us for the DVS.

In the following we will use the more general notation of an event as an indexed tuple containing the origin address k, a unique temporally ordered index i and creation time t which is expressed as a discrete 1us timestamp ts:

$$Ev_i = (k,t) \approx (k,ts) \tag{2.3}$$

For the dynamic vision sensor, the address index k is a function of coordinates of the pixels, the array dimensions (size x = sx and size y = sy) as well as its polarity pol which allows to express the event as a tuple of x, y, pol, ts:

$$pol = \begin{cases} 0 & \text{if } ON \text{ event} \\ 1 & \text{if } OFF \text{ event} \end{cases} \tag{2.4a}$$

$$k(x,y,pol) = 2 * (y * sx + x) + pol \tag{2.4b}$$

$$Ev_i = (k,ts) = (x,y,pol,ts) \tag{2.4c}$$

To handle the events and access specific entries in the tuple, property operators are defined:

$$Addr(Ev_i) = k \tag{2.5a}$$

$$Ts(Ev_i) = ts \tag{2.5b}$$

$$X(Ev_i) = x \tag{2.5c}$$

$$Y(Ev_i) = y \tag{2.5d}$$

$$Pol(Ev_i) = pol \tag{2.5e}$$

$$Sign(Ev_i) = \begin{cases} -1 & , Pol(Ev_i) = 0 \\ 1 & , Pol(Ev_i) = 1 \end{cases} \tag{2.5f}$$

This notation allows to easily define sets of events and perform operations or mutations of their properties.

2.4.1.2 Event generation

Another aspect when describing event-based systems which is in many publications not formalized is the neuron output function and event generation. In the following the output of a neuron with address k at time t will be described using an output function O[12]:

$$O(t)^k = \begin{cases} \{\} & \text{if } \neg \text{ event} \\ (k,t)_i, i^{++} & \text{if event} \end{cases} \quad (2.6)$$

Most events are encoding physical entities with the event criterion but this information is not part of the event itself but depends on the event's address. To simplify expressions which relate to these encoded entities P such as absolute light intensity, the $meaning$ function M is used as a concept to interpret events.

$$M(Ev_i) = P \quad (2.7)$$

where P is the physical entity encoded by the event criterion. This function is a conceptual construct that allows for instance to express the amount of change that is encoded by an event. Further examples can be found in the next chapter.

2.4.1.3 Indexing

Each event carries a unique temporal index i (as defined above) but in many algorithms, only events from a certain address k are of interest. The expression Ev^k is therefore used for any events with address k without the polarity bit:

$$Ev^k = \{Ev : Addr(Ev) = k\} \quad (2.8)$$

To allow a continuous ordering of the events from a specific event address, the subscript index j is used. While the index i is unique in an event stream, j is only unique for an address. The index j guarantees:

$$Ev_j^k \in Evs^k \quad (2.9a)$$

$$T(Ev_{j+1}^k) > T(Ev_j^k) \quad (2.9b)$$

$$Ev_j^k \neq \{\} \quad (2.9c)$$

[12] ($i^{++} = i \rightarrow i+1$)

To keep the formulations in this thesis consistent, cited equations from prior publications have been adapted from the original formulation.

2.4.2 Event Encoding

The concept behind event-based computation is to encode information in temporally discrete pulses instead of regularly sampled values. Event-based system are also called time encoding machines (TEMs) which are explained in more detail in (Lazar et al. 2011). In principle an analog signal can be converted into events in many ways but two ways are used the most: intensity and change encoding.

2.4.2.1 Intensity Encoding

The analog input signal can be directly converted into events. By integrating the signal (often as a current I_{in}) up to a threshold, registering the event, resetting the integration and thereby encoding the signal as the integration time.

Event Density Modulation

For a consistent naming within the field of event-based machine vision, this coding scheme is called event density modulation (EDM) but it is also known as pulse density modulation (PDM) , inter-spike interval (ISI) or rate coding in the context of neuroscience.

The simplest form of EDM coding is an integrate and fire neuron with a membrane voltage V_m, an input current I_{in}, a membrane capacitance C_{mem} to integrate the signal, a reset potential V_r and a firing threshold θ, the according equations are the following:

$$\frac{dV_m(t)^k}{dt} = \frac{I_{in}^k}{C_{mem}^k} \quad (2.10a)$$

$$V_m(t)^k = \begin{cases} V_m(t)^k \rightarrow V_m(t)^k & \text{if } V_m(t)^k < \theta \\ V_m(t)^k \rightarrow V_r & \text{if } V_m(t)^k >= \theta \end{cases} \quad (2.10b)$$

$$O(t)^k = \begin{cases} \{\} & \text{if } V_m(t)^k < \theta \\ (k,t)_i, i^{++} & \text{if } V_m(t)^k >= \theta \end{cases} \quad (2.10c)$$

For a constant input I_{in}, the event output frequency f_O which encodes the signal S_e can be computed by:

$$S_e = f_O = \frac{I_{in}}{C^k_{mem} \cdot (\theta - V_r)} \tag{2.11}$$

The variables of such an event-based system can of course also be replaced by other entities than physical circuit elements such as floating point variables or any other system capable of performing integration and thresholding. Even buckets that are filled with water can implement such a system (*Hydroneuron* 2012) but for a better understanding of the circuits in the following chapter, the entities are represented so that they can be implemented in electronic circuits.

Time to First Event

A variation of this encoding scheme, the time-to first event (TFE) encoding scheme also known as time-to-first spike (TFS) encoding scheme performs the membrane reset not after the generation of an event but on an external reset pulse P_r:

$$\frac{dV_m(t)^k}{dt} = \frac{I^k_{in}}{C^k_{mem}} \tag{2.12a}$$

$$V_m(t)^k = \begin{cases} V_m(t)^k \rightarrow V_m(t)^k & \text{if } P_r = 0 \\ V_m(t)^k \rightarrow V_r & \text{if } P_r = 1 \end{cases} \tag{2.12b}$$

$$O(t) = \begin{cases} \{\} & \text{if } V_m(t)^k < \theta \\ (k,t)_i, i^{++} & \text{if } V_m(t)^k >= \theta \end{cases} \tag{2.12c}$$

The signal S_e is thereby encoded by the time δT_e between the reset pulse and the event generation which can for a constant current I_{in} be computed by

$$S_e = \frac{1}{\delta T_e} = \frac{I_{in}}{C^k_{mem} \cdot (\theta - V_r)} \tag{2.13}$$

The actual encoding of EDM and TFE is the same and only the relative event timing and the number of events generated for a certain measurement differ.

2.4.2.2 Change Encoding

Encoding an analog signal as events has the advantage that the time domain onto which the signal is projected has a wide swing i.e. the integration time is not limited and therefore even small signals can be captured which leads to a wide dynamic range.

The drawback of intensity encoding is that it does not fully exploit the potential of events as discrete occurrences in time. Integration and thresholding generates artificial events which do not directly relate to temporal properties of the signal i.e. the phase of the events is less informative. The intensity encoding therefore also does not compress the information but it translates it into the time domain.

Change encoding on the other hand exploits the phase information of the events by representing a temporal property of the signal. If the change in signal is discretized, each event carries the information on when the signal changed by certain amount θ. By encoding changes, the events can remove temporal redundancies in the signal i.e. suppress output if the signal does not change by at least the threshold θ. This redundancy suppression leads to a compression of the signal and thereby to more efficient processing systems. To allow a partial reconstruction of the signal, the polarity of the change has to be encoded in some form. The simplest way of doing so is to use different addresses for positive and negative changes. In analogy to the nervous systems, these channels are usually called ON for the positive and OFF for the negative changes. To compute the change, the input I_{in} has to be sampled on each event generation I^i_{samp}. For a higher sensitivity, the change can be amplified by a constant factor a which leads to following description:

$$V_m(t)^k = a(I^k_{in} - I^k_{smpl}) \tag{2.14a}$$

$$O(t) = \begin{cases} \{\} & \text{if } V_m(t)^k < \theta \\ (2k+0,t)_i, i^{++} & \text{if } V_m(t)^k >= \theta \\ (2k+1,t)_i, i^{++} & \text{if } V_m(t)^k <= -\theta \end{cases} \tag{2.14b}$$

$$I^k_{smpl} = \begin{cases} I^k_{smpl} & \text{if } -\theta < V_m(t)^k < \theta \\ I^k_{last} & \text{if } V_m(t)^k <= -\theta \\ I^k_{last} & \text{if } V_m(t)^k >= \theta \end{cases} \quad (2.14c)$$

To encode for ON and OFF, the address k of the cell is broadened by an extra bit (LSB) indicating the polarity, ON = 0 and OFF = 1.
One drawback of encoding the change by events, is that the wide dynamic range of the direct encoding scheme is lost. To cope with this reduction in dynamic range, the input signal can be log-compressed which can be described as follows (the rest being the same as in Eg.2.14):

$$V_m(t)^k = a(log(I^k_{in}) - log(I^k_{smpl})) \quad (2.15)$$

2.4.3 Event vs. Sample

The topology and computations of the retina have evolved over millions of years and they perform a highly optimized way of visual encoding and processing (a view which is nicely summarized for neuromorphic engineering in chapter 3 of (Boahen 1997)). But the fact that visual signals are encoded in the form of events in most biological vision systems does not necessarily imply that this is also more efficient in artificial vision systems or machine vision applications.
To assess the suitability of event-based encoding for specific tasks, simple estimations can be performed. The output data rate D_O (bits/s) and the temporal resolution R_t (s) of a vision sensor array with dimensions w and h can be expressed depending on the encoding scheme and compared against sample based encoding. These estimations only concern the efficiency of encoding information and do not take into account who the actual encoding is implemented which would affect the size, prize, power consumption and figures alike.
In the following such a comparison is performed to assess and motivate the use of event-based encoding schemes in the field of machine vision. The figures are compared with a sample-based sensor which is in the case of a vision sensor a conventional frame-based imager: per frame each pixel gets sampled once.

Sample-based (Frame-based)
The output data rate and the temporal resolution in a conventional frame-based imager can be described as:

$$D_O = w * h * f_S * R_S \quad (2.16)$$

$$R_t = \frac{1}{f_S} \quad (2.17)$$

where the output data rate grows proportionally with the sampling rate f_S and the intensity (ADC) resolution R_S in bits. For a VGA image sensor ($w * h$ = 640x480) with a 10bit ADC running at f_S = 30Hz, D_O results in 92.2Mb/s = 11.5MB/s. The temporal resolution R_t is determined by the sampling rate which leads to a trade-off between the data rate and the temporal resolution.

EDM
In the event density modulation encoding scheme the intensity is encoded in the time between two events. To store this intensity information, the time between events has to be measured by a counter and stored as a relative timestamp. While the intensity resolution is determined by the counter frequency, the dynamic range is given by the longest exposure i.e. the biggest counter value that can be stored and communicated. In addition to this timestamp, each event also has to encode its address k which requires $size(k)$ bits. The EDM encoding figures can thereby be computed from:

$$D_O = w * h * \bar{f}_{EDM} * (S_{count} + size(k)) \quad (2.18a)$$

$$\bar{f}_{EDM} = \frac{\bar{I}_{in}}{C^k_{mem} \cdot (\theta - V_r)} \quad (2.18b)$$

$$R_t = \frac{1}{f_{EDM}} \quad (2.19)$$

$$size(k) = ceil(log2(w)) + ceil(log2(h)) \quad (2.20)$$

The output data rate grows proportionally with the average event rate $\bar{f}_{EDM}$ and counter size S_{count}. While the counter size S_{count} is constant, f_{EDM} grows linearly with the average light intensity. An integration current of 500fA under room light conditions, an integration range

$\theta - V_r$ of 1V and a capacitance C_{mem} of 20fF results in an event frequency of 25Hz. For a VGA ($size(k)$ =19) EDM sensor with a 16bit counter size (higher dynamic range than sample based) this results in 192Mb/s = 24MB/s. But the long exposure times for dark pixels (the average is 40ms but dark pixels have to integrate for much longer) will lead to motion blur and even more important: if the same sensor is used outdoors under sunny daylight conditions, the data rate grows by a factor of 100 to 2.4GB/s. The temporal resolution is also a function of the light conditions and while fast movements can easily be captured in bright light, they cannot be seen in the output of the sensor under darker light conditions.

TFE

The time-to-first-event encoding is similar to the EDM encoding:

$$D_O = w * h * f_S * S_{count} \tag{2.21}$$

$$R_t \geq \frac{1}{f_S} \tag{2.22}$$

The output data rate grows proportionally with the sample rate f_S (reset rate) and the counter size S_{count}. This combines the fixed data rate of the sample-based approach with the high dynamic range of the EDM approach. One advantage of a regular reset frequency is that the timestamps can be stored in memory before communicated so that the full frame is transmitted at once and the event addresses do not have be stored. If the counter size is 16 bit, this results in a data rate of 147.5Mb/s = 18.4MB/s for a reset frequency f_R = 30Hz.
The lower bound of the temporal resolution is given by the reset rate but this does not guarantee that dark pixels can finish their integration during this time so that dark pixels do not produce a measurement. Another major drawback is that events from uniformly illuminated part of the scene are communicated at the same time which leads to imprecisions in the timing measurement and thereby resolution (Liu et al. 2014).

Change

In the change encoding scheme the figures can be computed in the following way:

$$D_O = w * h * \bar{f}_{epps} * size(k) \tag{2.23}$$

$$R_t = \frac{1}{f_{cutoff}} \tag{2.24}$$

The data rate of the change encoding event-based image sensor is dependent on the events per pixel per second $\bar{f}_{epps}$ and the address size of k. $size(k)$ is calculated from the bits required for x and y address plus 1 bit for polarity:

$$size(k) = ceil(log2(w)) + ceil(log2(h)) + 1 \tag{2.25}$$

The event frequency depends on visual change. Estimates of epps and data rate from a dynamic vision sensor with about 15% constrast sensitivity (DVS128, (Lichtsteiner et al. 2008)) calculated for a VGA sensor (32bit assumed as $size(k)$[13]).
The temporal resolution is given by the cutoff frequency f_{cutoff} of the sensor i.e. the time it takes to detect and communicate a change. This cutoff frequency can be a function of the illumination, the biasing as well as temperature.

Table 2.1 Exemplary events per pixel per second values (epps) and data rates upscaled to vga from DVS128 recordings. **POV=point of view. Data from jAER Wiki.

Scene	epps (Hz)	D_O VGA
Walking (POV**)	10	12.2MB/s
Moving Hand	6.3	7.6MB/s
Car Driving (POV)	2.2	2.8MB/s
Cars & People	0.2	224kB/s

Comparison

The considerations on data rate and temporal resolution show clearly that a sample-based approach is suitable if both figures have to be fixed and cannot be a function of the input. Otherwise the change encoding outperforms the sample-based encoding in data rate as well as temporal

[13] In the real DVS the 16bit used for the timestamp would have to be added

resolution. These insights into the encoding efficiency of the change event-based encoding are one of the motivations the advancement of the research in the field of event-based machine vision.

While TFE limits the data rate compared to EDM, it also constrains the dynamic range of the pixel by aborting the integration time. The fact that the response of EDM and TFE are both dependent on absolute light intensities is unfavorable and they might only be employed where a high dynamic range is really necessary.

2.5 Definition: Event-Based Machine Vision

The field of event-based machine vision (EBMV) can be defined as follows:

Event-based machine vision is an engineering discipline which solves visual problems such as the localization, recognition and description of objects using sensors that encode information in the form of independently created, temporally discrete events.

For the full understanding of this definition certain expressions are explained in detail:

- *engineering discipline*: EBMV does not have the goal of developing bio-inspired systems even though nature might be used as inspiration for certain processing procedures. The goal of this field is solve engineering problems in a more efficient way than by frame-based machine vision.
- *visual problems*: This term points to all problems which can be solved using vision i.e. using light and a multitude of light-sensitive elements. The term *visual* is thereby not an attribute of the problem but of its solution. *Problem* refers to all sorts of challenges which serve a market demand when solved. Being an engineering discipline, as described above, the social relevance determines the importance of a problem and intellectual puzzles are not considered problems. The list of stated examples of such visual problems namely localization, recognition and description is not complete.
- *encode information*: The process of translating light intensity into another form. In most cases the *encoding* is a conversion of the light into some sort of analog or digital voltage signal.
- *independently created*: This expression can be considered to be the most important in the definition above because it differentiates event-based from frame-based machine vision. *Independently* in this context means without the requirement of an external sampling signal i.e. an external command that instructs the pixel to perform the encoding of the visual information. The absence of an external sampling signal inverts the processing chain so that it is the sensor dictating when to process information and not the downstream processing units.
- *temporally discrete events*: For *events* it is very important that they have a clear onset in time; there is no gradual *event* because either it is there or not. It is important to mention that the events do not necessarily have to be created or communicated asynchronously.

The following chapters will focus on these stated sensors, visual problems and solutions.

2.5.1 Frame-Free Digital Vision

Most of the work reviewed and reported in this thesis is based on the research of T. Delbruck et al. who termed the field *Frame-Free Dynamic Digital Vision*(Delbruck 2008; Delbruck 2012). The term *Event-Based Machine Vision* used in this thesis tries to capture the field more systematically: The field is a sub-field machine vision because it tackles the same problems but specifically with data that is based on events as underlying structure. Using *Event-Based* instead of *Frame-Free* allows to establish a link to other fields which process sensory information such as event-based machine audition whereas it might be less sensible to speak of frame-free machine audition. While dynamic vision characterizes the sensor modality of the DVS128 very well, the

field should be capable of also including sensors and algorithms which use events that do not necessarily encode the dynamics in a scene. And since the problems of the field overlap with the ones in machine vision, it makes sense to refer to this field instead of the rarely used *Digital Vision*. The expression *Event-Based* was preferred over *Event-Driven* which is also used in some publications because in many cases the computation is still clock-driven and the events are simply used as a way of encoding information.

3 Event-Based Vision Sensors

All event-based machine vision starts with a sensor that converts a visual scene into a stream of events. This chapter gives an overview over the different types of event-based vision sensors and their characteristics. Most sensors are described only briefly but for the dynamic vision sensor and its successors a more detailed discussion is given.
The sensors can be classified according to the input modality to which they react which gives the basic structures for the sections in this chapter. The chapter first explains how conventional imagers work and gives an overview over the existing event-based vision sensors (many of them also summarized in (Delbruck et al. 2010c; Posch et al. 2014; Liu et al. 2014)). The implementation and results of the dynamic and active pixel vision sensor (DAVIS) are reported in a separate section.

3.1 Conventional Imagers

Nowadays most vision sensors are conventional, frame-based image sensors (also known as imagers). They consist of a 2D array of pixels which sense the light intensity. Each pixel contains a photo-sensitive element to convert the incoming photons into an electronic signal. Most imagers can be classified according to two classes: CCD or CMOS imagers.

3.1.1 Charge-Coupled Device (CCD) Imagers

Originally developed as a memory technology, charge-coupled devices (CCDs) allow to store and move charges in a semiconductor using electric fields. Since these devices are also photosensitive, they are also used as electronic imagers that accumulate charges during the exposure which are then moved to the periphery where they are converted into an electronic signal. CCD cameras dominated the camera market in the beginning because they reliably produced images of good quality but they have a major drawback: The processes in which they are fabricated are optimized for charge generation and transportation which makes it hard to integrate other circuits on the imager.

3.1.2 CMOS Image Sensors (CIS)

Complementary metal-oxide-semiconductor image sensors (CMOS imagers) use a production processes derived directly from the one used in regular computer chips and therefore they allow the integration of other components such as fast, parallel analog to digital converters and synchronous logic. Since charges cannot be moved easily through such chips, the signals of the pixels have to be amplified before they can be read out. The transistors required for this readout cannot be matched precisely and in the beginning of the CIS technology, mismatch was too large to compete with CCD imagers. With the advancement of the CMOS fabrication process, the transistors became smaller and noise cancellation techniques such as correlated double sampling (CDS) allowed CIS to catch up with CCD image quality but at a lower price. For these reasons most webcams, smartphone and tablet cameras and an increasing number of digital single-lens reflex (DSLR) cameras nowadays are based on CMOS image sensors.

3.1.2.1 Active Pixel Sensor (APS)

In contrast to CCD, CMOS technology does not allow to move charges from a pixel to the periphery without adding significant noise. Therefore, the first generation of CMOS image sensor using passive pixels suffered from strong noise. The concept behind the active pixel sensor (APS) is to buffer the signal within the pixel before reading it out.
Fig.3.1 shows the schematics of a so called 3T APS pixel: The reset transistor (MN1) charges the parasitic capacitance of the photodiode at the beginning of a readout cycle. The photocurrent

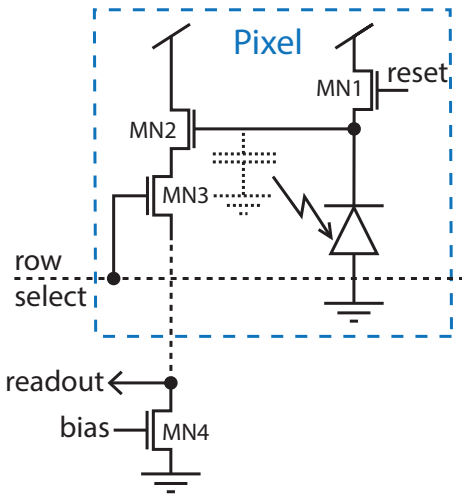


Figure 3.1: 3 Transistor (3T) active pixels sensor (APS) pixel schematic.

generated by the photodiode then discharges this capacitance during the exposure time and at the end of the readout cycle, the remaining charge is read out. This amplified read out is performed by the source follower circuit formed by MN2 which generates a current from the integrated voltage and MN4 which converts this current into a voltage again ("readout" in Fig.3.1). To assure that only one row can "write" its current to the readout line shared by all pixels in a column, a select transistor MN3 is required.

3.1.2.2 Noise in CMOS image sensors

A key problem with the APS pixel is the mismatch among the transistors within the pixels because the same amount of light integrated over a fixed time window can lead to different readout voltages from different pixels. The variations in the readout voltage i.e. the pixel noise has a temporal and a spatial component: If the pixel gets exposed over and over again, their response can be averaged and the temporal noise is removed while the spatial component, called the fixed pattern noise remains. In the following paragraphs the different components are discussed.

Fixed Pattern Noise (FPN)
The offset component of the fixed spatial noise is better known as fixed pattern noise (FPN). Due to mismatch in the pixels as well as in the readout circuits, the same amount of integrated charge leads to different readout voltages. The dark signal non-uniformity (DSNU) is the pixel response offset from the average pixel response in the absence of external illumination. Before the introduction of correlated double sampling schemes, this was the dominant noise source that obstructed the success of the CMOS technology. Since the readout in the APS pixel uses a column-shared transistor (MN4 in Fig.3.1), mismatch in this transistor leads to stripe patterns in the images.

Photo Response Non-Uniformity (PRNU)
The gain component of the fixed spatial noise is better known as photo response non-uniformity (PRNU). This non-uniformity is caused by the mismatch in the readout transistors and the integration capacitance which lead to different conversion gains from photo electrons to readout voltage. This mismatch becomes more pronounced for longer exposures but is generally weaker than the offset mismatch.

Shot Noise
The photon shot noise is a consequence of the fact that the number of electrons generated by the incoming light follows a Poisson probability distribution. The uncertainty for a given electron count following such a distribution i.e. the shot noise is the square root of the signal $\sqrt{S}$.

Reset / kTC Noise
The thermal kTC or reset noise is a consequence of the fact that when the pixel gets reset, the current fluctuates due to thermal motion of the electrons. When this fluctuating current is sampled on a capacitance C, the according sampling noise voltage is proportional to $\sqrt{\frac{kT}{C}}$.

Flicker / 1/f Noise
The $1/f$ noise is caused by charges trapped in the readout transistor that lead to current fluctuations with a power density profile that drops

off with $1/f$ down to the point where to fluctuations are dominated by the thermal noise.

3.1.2.3 Correlated Double Sampling (CDS)

A effective method to remove FPN as well as the kTC sampling noise is correlated double sampling. After the pixel is reset, this reset voltage is sampled and stored on a sample-and-hold capacitance. After the exposure, the actual signal is subtracted from this first read. Since the reset and the signal voltage are stored on the same capacitance this circuit has a low common-mode noise rejection but in return it suppresses kTC noise. Another disadvantage of the circuit is the fact, that the capacitance-mediated voltage subtraction reduces the readout gain (Nakamura 2006).

Differential Delta Sampling (DDS)
Differential delta sampling works similar to CDS but instead of sampling the two signals on the same capacitance, they are stored separately and the signal difference is amplified. While this approach removes the drawbacks of the CDS, the double sampling of the signal leads to more kTC noise (Nakamura 2006).

3.1.2.4 Rolling Shutter Effect

To improve the readout speed and to lower the price with a simple pixel design, many CMOS image sensors use a rolling shutter. Instead of exposing all pixels at the same time, they are exposed and read out row by row. This non-uniform exposure leads to artifacts in the image called rolling shutter effect: Moving objects are exposed at different times for different rows and therefore they are distorted.

3.2 Intensity Sensors

Some of the early sensors that used events as their output directly measured light intensity. As discussed in the previous chapter, event-based intensity measurements can either be by event density modulation or time-to-first event encoding.

3.2.1 Event Density Modulation Imagers

The concept of translating light intensity at a pixel into event frequency which is called event density modulation (EDM) is also known as pulse density modulation (PDM), inter-spike interval imaging (ISI) or as octopus silicon retina because of the similarity to it's biological counterpart. The advantages of this type of sensing are its wide dynamic range because the integration time is not limited and the simple and cheap analog-to-digital conversion: the inverse of the inter-event interval corresponds to the light intensity. The first imagers of this kind were described in 1995 by Mortara (Mortara et al. 1995) and later rediscovered by Culurciello and colleagues (Culurciello et al. 2001b; Culurciello et al. 2001a; Culurciello et al. 2003) and improved (Culurciello et al. 2004). A similar circuit was also used in a foveated imager that contains static and dynamic pixels (Azadmehr et al. 2005). All three versions of this sort of imager design suffered from strong fixed pattern noise because they did not employ a correlated double sampling (CDS) scheme or something alike. By using a feedback capacitance, a similar effect to CDS can be achieved and the fixed pattern noise lowered (Olsson et al. 2008a). Using a stacked photodiode allows to get color information from a single pixel (Olsson et al. 2008b; Olsson et al. 2009).
The basic principle of the pixel is to charge up a capacitance C_m which is then discharged by the photocurrent I_p. If the voltage V_m on the capacitance reaches a fixed threshold θ, an event is generated and the pixel is reset i.e. the capacitance is recharged to the reset potential V_r. This leads to a spiking frequency which is proportional to the photocurrent. The response of these imagers can be described as:

$$\frac{dV_p(t)^k}{dt} = \frac{I_{ph}^k}{C_p} \tag{3.1a}$$

$$V_m(t)^k = \begin{cases} V_m(t)^k \rightarrow V_m(t)^k & \text{if } V_m(t)^k < \theta \\ V_m(t)^k \rightarrow V_r & \text{if } V_m(t)^k >= \theta \end{cases} \tag{3.1b}$$

$$O(t)^k = \begin{cases} \{\} & \text{if } V_m(t)^k < \theta \\ (k,t)_i, i \rightarrow i+1 & \text{if } V_m(t)^k >= \theta \end{cases} \tag{3.1c}$$

EDM imagers exhibit a wide dynamic range because the intensity signal is not mapped onto a fixed voltage range where the dynamic range is limited by the exposure time but by a time window which can be arbitrarily long. This readout in the time domain also comes with drawbacks: The non-uniform exposure time can lead to severe motion artifacts. Bright pixels produce many events and thereby increase the power consumption and occupy the output bandwidth. When the output bus is occupied, it introduces imprecision in the timing of the other events which should be transmitted and while this error can be averaged away for the bright pixels which generate plenty of measurements, this is not possible for dark pixels which suffer more from the occupied bus. This issue can be addressed using a time-to-first-event (TFE) imager scheme.

3.2.2 Time to First Event Imagers

While the EDM imagers allocate more output bandwidth to bright pixels, this is avoided in Time to First Event (TFE) imagers where all pixels are reset concurrently through an external signal (Guo et al. 2007; Shoushun et al. 2007). The advantage of this scheme is that it preserves the high dynamic range of the EDM while guaranteeing that each pixel only produces one event. The drawback is that all pixels of the same intensity will generate events at the same time and uniform parts generate a temporal bottleneck which leads to timing imprecision.

3.3 Spatial and Spatiotemporal Contrast Sensor

With the intention of modeling the outer plexiform layer (OPL) of the retina, Boahen developed the first event-based sensor that integrated spatial information at a single pixel (Boahen 1996a; Boahen 1996c; Boahen 1997). The photocurrent is spread using two reciprocally connected resistive networks that form a spatiotemporal bandpass filter with local automatic gain control. The output signal of this OPL model is then fed into a circuit that converts it into a event frequency similar to the EDM imagers except that the quantization is adaptive. This concept of event-based spatiotemporal vision sensor was then further developed and resulted in a sensor with spatial and temporal contrast output (Zaghloul et al. 2004b; Zaghloul et al. 2004a; Zaghloul et al. 2006) which is also mentioned in Boahen's TED Talk (Boahen 2007)

While the silicon retina approach to spatial contrast is operating in continuous time and the orientation of the contrast is not known, Barbaro et al. proposed to compute the temporal contrast magnitude and orientation by comparing a pixel's intensity with its neighbors along the x- and y-direction in a TFE fashion (Barbaro et al. 2002). The sensor contains two output channels: one encodes the contrast intensity as time to first event and the other one encodes the orientation as event time relative to a sinusoidal reference signal which is sent to all pixels. This sensor design was then improved in dynamic range and sensitivity and made use of a voltage representation which is independent of the illumination level (Ruedi et al. 2003).

A central problem when comparing light intensities between pixels as voltages or currents is mismatch in the transistors used, which does not allow to copy a current perfectly. This mismatch leads to constant offsets among pixels and thereby to noise. A possible solution to get rid of this mismatch is to calibrate it away using on-pixel digital to analog converters (DACs) (Costas-Santos et al. 2007). While (Costas-Santos et

al. 2007) delivers unsigned contrast information that does not tell whether a pixel is brighter than average, an improved version delivers signed spatial contrast (Lenero-Bardallo et al. 2010; Leñero-Bardallo et al. 2010). The large pixel size of these pixels and sensors limit its applicability.
In addition to these realized designs, there has also been a suggestion for a design using order-based coding (Thorpe et al. 2010).

3.4 Temporal Contrast Sensors

Already the first event-based vision sensor measured temporal contrast (Mahowald 1992): The photocurrent is converted into a logarithmic voltage and compared to low-pass filtered version of this voltage and as soon as this difference exceeded a fixed threshold, the pixel generates an event. Ten years later the field was revived when the late Jörg Kramer developed temporal contrast circuits with a better performance (Kramer 2002a) which eventually resulted in what can be considered one of the predecessors of the dynamic vision sensor (Kramer 2002b): the ON/OFF transient imager (OOTI). Independent of this work, Azadhmer et al. developed their own event-based change detection circuits further discussed under "Dual Readout Sensors".

3.4.1 Dynamic Vision Sensors (DVS)

The OOTI used a low-pass feedback circuit to generate a voltage proportional to the time derivative log-brightness on the pixel. The current flowing onto and from this feedback node is used in two independent branches to compute whether it exceeds an upper or lower threshold and to create an ON event (or OFF event respectively) if this is the case. This circuit suffers from two main problems: it has a strong mismatch in the thresholds for generating events which makes it hard to compare the output of different pixels and it does not integrate the change. Its response is dependent on the slope of the temporal contrast and not the absolute amounts of change i.e. it might not create events if an object is moving slowly.
P. Lichtsteiner and T. Delbruck then improved this concept by redesigning multiple aspects (Lichtsteiner et al. 2004; Lichtsteiner 2006): The photodiode capacitance was reduced to increase the photoreceptor bandwidth, the ON and OFF thresholds were separated, the critical transistors and capacitors re-sized. When this improved version of the OOTI was published, P. Lichtsteiner and T. Delbruck were already working on a new way of sensing temporal contrast: The dynamic vision sensor (DVS). In contrast to its predecessor, the dynamic vision sensor performs its computations in the voltage domain which reduces the mismatch and noise because the ON and OFF currents do not have to be amplified with current mirrors. It also decouples the different computational parts of the system into dedicated sub-circuits to avoid oscillations: logarithmic photoreceptor, buffer, amplified differentiator, threshold comparison and event communication (Lichtsteiner et al. 2010). The first DVS chip was built for the CAVIAR project and contained a pixel array of 64x64 pixels (Lichtsteiner et al. 2005), a second version was a line sensor optimized for high-speed applications (Lichtsteiner et al. 2006a) and the best known version contains an array of 128x128 pixels as well as an on-chip bias current generator (Lichtsteiner et al. 2006b; Lichtsteiner et al. 2008) (also known as Tmpdiff128). Recently a low-power version of the sensor was developed (Berner et al. 2014). Because of its importance in the field of event-based machine vision and the presented work, the DVS is discussed in more detail in the following.

3.4.1.1 DVS Pixel

Fig.3.3 shows the basic blocks of the DVS pixel (the detailed schematics can be found in Fig.3.4): the photodiode converts the light at the pixel into a current I which is supplied by the transistor above (M_{fb} in Fig. 3.4). By feeding back the photoreceptor voltage to the gate, the photoreceptor voltage is clamped and the gate voltage

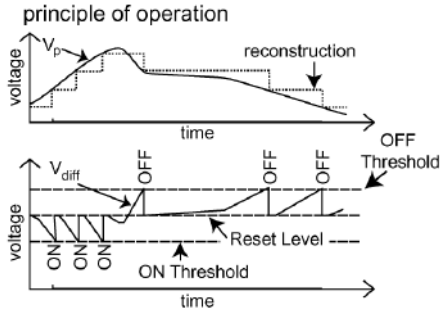


Figure 3.2: Dynamic vision sensor principle. Reprinted from (Lichtsteiner et al. 2008).

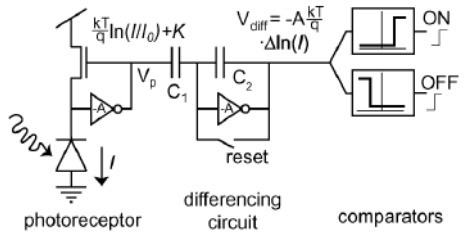


Figure 3.3: Dynamic vision sensor pixel schematic. Reprinted from (Lichtsteiner et al. 2008).

becomes a logarithmic function of the photocurrent (Delbruck et al. 1994). This logarithmic compression makes the pixel follow the Weber-Fechner law which states that the just noticeable difference in a stimulus is normalized by the absolute stimulus intensity. This makes the contrast perception a logarithmic function of the stimulus intensity. Apart from this biological background, the logarithmic compression allows the pixel to span a wide dynamic range of photocurrents without getting saturated.

The logarithmic voltage then gets buffered (not shown in Fig. 3.3) using a source follower circuit (M_{b2} and M_{sf} in Fig. 3.4) to charge the capacitor C_1. The other side of this capacitor, a floating node, is attached so that voltage deviations since the last reset get inversely amplified through capacitor C_2 by the ratio $A = C_1/C_2$. Using capacitors as the differencing circuit removes the offset including the offset caused by mismatch and since capacitors can be better matched than transistors, it also reduces the gain mismatch. The outputs of the differencing stage V_{diff} is then fed into two current comparators which compare it against a lower "ON" and an upper "OFF" threshold. The OFF signal has to be inverted because the OFF comparator generates an active-low signal.

As soon as the temporal contrast exceeds one of the two thresholds, it is used to generate an event i.e. an address event handshake. The outputs of the comparators are used as a pull-down signal on a row request line which initiates the AER handshake described below. After the column handshake is also acknowledged, the pixel gets reset by shorting the differencing circuit feedback, thus balancing the amplifier at its reset level. As shown in Fig.3.2, increases in V_p lead to decreases in V_{diff} down to the point where they exceed a lower threshold and initiate the event communication and thereby the pixel reset.

3.4.1.2 AER

For the communication of an event, the DVS uses the address event representation protocol (AER). Upon a threshold crossing the row request RR is pulled down and the four phase handshake gets initiated as shown in Fig.3.5b. As events might occur concurrently in different rows, the row requests have to be arbitrated by a row arbiter circuit (see Fig.3.5a) and one of these lines gets chosen randomly and acknowledged by raising the corresponding row acknowledge signal RA. At the same time the handshake logic of this row writes the row's address onto the address output bus. The row acknowledge allows the pixel to pull down the request line in the column direction by pulling down $CRON$ or $CROFF$ which is also arbitrated, acknowledged and encoded. The column acknowledge CA not only resets the pixel but also leads to a chip request REQ. As soon as the 15 bit (7x + 7y + 1polarity) is registered in the periphery, the chip acknowledge is raised and the row and column requests are released to release the chip request and allow the periphery to release the

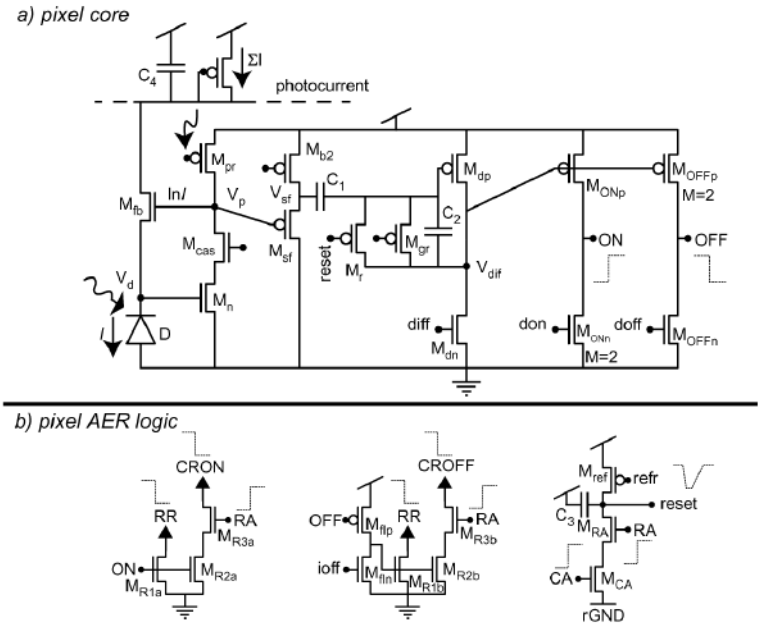


Figure 3.4: Detailed schematics of the DVS pixel: a) transistor level schematic of fig. 3.3 b) asynchronous AER circuits. Reprinted from (Lichtsteiner et al. 2008).

ACK which allows to begin with the transmission of another event.

3.4.1.3 Bias Current Generator

Many parameters in the pixel as well as in the periphery are set by bias currents (such as the ON or OFF threshold). The gate voltages to generate these currents were initially supplied by potentiometers but this was tedious to set up, hard to reproduce, troublesome to modify in operation and most importantly the currents changed with the temperature of the chip and across dies because of threshold and supply voltage variations. For this reason T. Delbruck and P. Lichtsteiner developed a fully programmable bias current generator that allows to digitally configure the desired current (Delbruck et al. 2006). The idea behind this current generator is to sequentially split a master reference current and program which of these split currents are summed up to generate an output current which is copied and buffered to the gate that has to be configured. In comparison to a linear DAC, this approach has the advantage that the voltages generated allow to program even small differences in current while the linear resolution of a DAC does not have enough detail on the voltages of interest.

3.4.1.4 Vision Sensor

The Tmpdiff128 chip is interfaced to a complex programmable logic device (CPLD) that receives the chip request as well as the event address. If the chip request is raised, the address gets registered as a 16 bit word (1 bit to denote the type of the word) and timestamped with another 16 bit word (14bits for the actual timestamp). The timestamp has a resolution of 1us and since a counter with only 14 bits has to restart every 16.384 ms, so-called wrap events are used to indicate such a counter overflow and to allow a timestamp expansion up to 32 bit (>1h) on the host. The address and the timestamp are transferred to a FIFO buffer which is then trans-

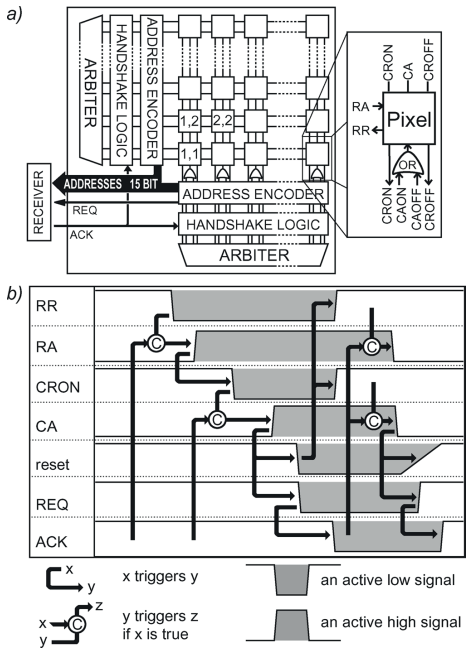


Figure 3.5: The Address Event Representation on the DVS: a) block level diagram of the communication circuits and b) timing diagram of the event communication. Reprinted from (Lichtsteiner et al. 2008).

ferred through a Cypress FX2 chip to the host using the USB 2.0 protocol.

3.4.1.5 Characterization

Since every pixel processes information in continuous time without the requirement of an external clock, events can be communicated without the requirement for a scanner which leads to a latency of down to 15us (for bright conditions). The sparse representation and the absence of analog-to-digital converters (ADCs) lead to a low power consumption of 30mW. Since the pixel's operation does not depend on a globally fixed exposure time, the in-scene dynamic range can go up to 120dB.

3.4.2 Low Contrast DVS

The success and the potential of the dynamic vision sensor led to further research in the field. One key figure that was a target for improvement was the contrast sensitivity. (Delbruck et al. 2010b) increased the sensitivity by using two successive gain stages to amplify changes. (Lenero-Bardallo et al. 2011a) was also using a pre-amplifying stage in addition to a faster photoreceptor circuit. (Serrano-Gotarredona et al. 2013b) then further increased the sensitivity using trans-impedance pre-amplifiers.
Two key problems with low contrast DVS are: The increased sensitivity leads to more data, power consumption and temporal jitter and it also limits the intra-scene dynamic range (for (Lenero-Bardallo et al. 2011a; Serrano-Gotarredona et al. 2013b) it is a 60dB instead of 120dB).

3.4.3 Infrared DVS

To extend the application scenarios to low light scenes, the DVS technology was already very early on adapted to become IR sensitive by using bolometers (Matolin et al. 2008; Posch et al. 2008b; Wohlgenannt et al. 2008; Posch et al. 2009). The microbolometer translates heat into resistance and the DVS circuit then detects and amplifies changes in the current flowing through this resistance. In a second iteration the sensitivity of these IR sensors was increased by using a two-stage amplifier (Posch et al. 2010c).

3.4.4 Line DVS

Under constant lighting, the DVS is highly suited for high-speed line sensor setups. Line sensors only contain a few lines of pixels and if the speed of an object passing the sensor is known, the full image can be reconstructed. The second generation of DVS was such a dual line 2x64 pixel sensor (Lichtsteiner et al. 2006a) and also the Tmpdiff128 can be configured to operate as line sensor. In a next generation of the line sensor, the resolution was increased to 2x256 and an on-chip timestamp generator was included (Posch et al. 2007b; Posch et al. 2007a). If the line sensor is mounted on a rotating head as in (Belbachir

et al. 2010c) (which unfortunately lacks any reference to the DVS), it is possible to get 360° panoramic views. The latest development in the field of line sensors is a stereo system of 1024 pixel line sensors rotating at up to 10 revolutions per second that delivers spatial contrast images, intensity images and depth maps (Belbachir et al. 2014).

3.4.5 Synthetic DVS

In some applications it is beneficial to use an emulated DVS instead of real application specific integrated circuits (ASICs). These so called synthetic DVS which are based on conventional frame-based cameras allow for instance the exploration of new sensor designs. A first such system was developed by Paz-Vicente et al. (Paz-Vicente et al. 2009) and later on improved by using a cheap of-the-shelf PS3Eye image sensor (Katz et al. 2012).

3.5 Color Sensitive Sensors

The penetration depth of light in silicon depends on its wavelength: while blue is absorbed near the surface, red light can travel up to a few microns into the silicon before getting absorbed. Stacking photodiodes on top of each other, allows to infer the color of the light falling onto a pixel without the requirement of color filters that absorb 2/3 of the photons. This principle has first been used commercially in the Foveon X3 image sensors (Lyon et al. 2002).

3.5.1 Color EDM Sensors

The first event-based sensor using a stacked photodiode to infer color information was developed by Olsson and Hafliger (Olsson et al. 2008b; Olsson et al. 2009) and encoded the light intensity as inter-spike intervals. In the following this sensor was improved by adding a third photodiode (Lenero-Bardallo et al. 2011b; Lenero-Bardallo et al. 2012).

3.5.2 Color DVS

Berner et al. applied the stacked photodiode principle to the DVS technology (page 34) (Berner et al. 2010; Berner et al. 2011) but unfortunately the available standard CMOS photodiodes did not allow a satisfying color discrimination.

3.5.3 Color-Based Face Detection Sensor

The principle of stacked photodiodes is also used in a dedicated vision sensor for low-power face detection that compares the redness of the center with the surrounding pixels (Berner et al. 2008). The sensor also contains dedicated circuits to detect eye-like patterns i.e. pixels that are darker than its surround as well as EDM readout circuits. The color and eye information is integrated in a face detection circuit which is intended to allow powering up a device on the presence of a face.

3.6 Dual Readout Sensors

3.6.1 Asynchronous Time-Based Image Sensor (ATIS)

Based on the insights and ideas Posch gained when supporting the design of the DVS, he designed a new type of sensor that combines the temporal contrast readout with an event-based intensity readout. This asynchronous time-based image sensor (ATIS) has been somewhat repetitively covered by publications (Posch et al. 2008a; Matolin et al. 2009; Matolin et al. 2010; Posch et al. 2010b; Posch et al. 2010a; Posch et al. 2011a; Posch et al. 2011b).

The basis of this sensor is a pixel in which the creation of a DVS event triggers the intensity readout: A capacitor gets charged and then integrated using a photocurrent of a second photodiode. As soon as the voltage on the capacitor drops below an upper threshold, a first intensity event is triggered and as soon as it passes a lower threshold a second intensity event is generated. The time between these two events corresponds to the inverse of the light intensity.

The advantage of this intensity measure in the

time domain is a wide dynamic range of up to 143dB which is a result of the long exposure times used to capture dark parts of a scene. But the motion artifacts make this sensor not suitable for dynamic scenes or the observation of fast moving objects. The integration might also be interrupted by a new event and fast; thin objects can therefore become invisible in the intensity readout. In addition to these conceptual drawbacks, the ATIS requires two photodiodes and a complex, area-consuming readout circuit.

3.6.2 Dynamic and Active Pixel Vision Sensor (DAVIS)

R. Berner realized that the DVS circuit does not consume the photocurrent and that it could be reused for an intensity readout so he implemented a log intensity readout (Berner 2011; Berner et al. 2011). Even though this circuitry allowed to calibrate for offset and gain mismatch, the acquired images were very noisy and the calibration tedious. Therefore he decided to use a conventional APS readout circuit which lead to the dynamic and active pixel vision sensor (Berner et al. 2013b; Berner et al. 2013a; Brandli et al. 2014a). This sensor is extensively described in the according study section on page 41.

3.6.3 Other dual readout sensors

3.6.3.1 Foveated Intensity and Change Imager

The concept of foveated sensors i.e. the principle of using different pixels in the center of the imager than in the surround of the array, is inspired by the retina and in particular the fovea. Apart from sensors with conventional readout, Azadhmer et al. developed an event-based, foveated sensor: smaller EDM intensity pixels in the center and a design of temporal contrast pixels in the periphery that was developed independently from the work of J. Kramer and T. Delbruck (Azadmehr et al. 2005). The idea behind this sensor design is to use the motion cues in the periphery to steer the higher-resolved intensity pixels to the interesting parts of the scene.

3.6.3.2 Motion and Region of Interest Sensor

With a similar motivation as in the foveated event-based sensors, another design clusters DVS pixels together into motion super-pixels: the photocurrent of 4 photodiodes is sampled, differences amplified and capacitively coupled with the neighboring pixels and as soon as this difference exceeds the ON or OFF threshold, an event is generated (Zhang et al. 2012; Zhang et al. 2013). These asynchronous motion super-pixels indicate interesting regions and a synchronous scanner then allows to read out the buffered and the log-compressed photocurrent of each photodiode. While the mismatch in the motion detection is averaged away using the capacitive network, the intensity readout lacks the possibility do perform a double sampling scheme which leads to a considerable amount of noise.

3.6.3.3 Intensity and Spatial Contrast Sensor

Based on the EDM pixel, Lenero-Bardallo and Hafliger have developed a pixel circuit that allows subtracting the average intensity event density among its four neighbors from the intensity event density of a single photodiode and thereby compute the spatial contrast at the pixel. If this subtraction is turned off, the pixel performs EDM intensity encoding and when the AER reset of the pixel is turned off, it encodes the intensity or the contrast as TFE (Lenero-Bardallo et al. 2014).

3.7 Computing Pixel Sensors

Apart from just computing spatial or temporal contrast some other event-based sensors can compute more complex functions at the pixel level.

3.7.1 Programmable Pixel Sensors

While most of the sensors presented here perform their computation in continous time, this can also be done synchronized. Dudek et al. have implemented multiple generations of vision chips in which each pixel also contains a mixed signal processor with a reduced instruction set. This single instruction, multiple data (SIMD) architecture allows computation on the pixel by exchanging information with its neighbors and performs simple operations such as add subtract or halt (based on comparator output) (Dudek et al. 2001; Dudek 2005; Lopich et al. 2010; Lopich et al. 2011; Carey et al. 2013). Even though these sensors are clocked, they can still be considered event-based because the on-pixel processing allows to detect specific events which allows to perform redundancy suppression.

3.7.2 Object Tracking Sensors

Temporal contrast is an important cue to track objects (also used in the foveated or ROI sensors) and it is possible to cluster the temporal contrast output already on the image plane to determine the extent of a moving object. This sort of computation requires a dynamic rewiring and comparison of pixels with its neighbors. Combining these comparisons with a winner take all circuit allows to output a single position for an object i.e. cluster of pixels (Chan et al. 2007).

3.8 Study of DAVIS Sensor

The dynamic and active pixel vision sensor combines the advantages of the DVS with the possibilities of conventional frame based imagers on the pixel level by reusing the DVS photocurrent so that they two readout modalities do not interfere with each other. This section gives an overview over the different DAVIS designs which were produced and then discusses specific aspects in detail.

3.8.1 Chip generations

After the success of the DVS, the research group of T. Delbruck focused on improving the technology with the development of higher resolution sensors, a word-serial AER protocol and color sensitivity. But these developments took more time than expected because they were more challenging than assumed. The next chance to move the technology further with all of the experiences gathered came with the FP7 EU SeeBetter project which allocated money for multiple chip runs: 3 test runs, 2 larger array runs and a full wafer run.
The DAVIS technology was developed on these SeeBetter chip runs: the test runs SeeBetter10, SeeBetter11, SeeBetter20 (DAVIS 64a), the full array chips SBret10 (DAVIS 240a) and SBret20 (DAVIS 240b) and a full wafer run (SeeBetter wafer) containing DAVIS 128a, DAVIS Sense 192a, DAVIS 240c, DAVIS 346a, DAVIS 346b, DAVIS BSIa, DAVIS RGBa, DAVIS 640a as well as an experimental event-filter design and 2 cochlea designs. Since the circuits on all of these chips are all very similar, the following overview is only superficial. The first name in the title of a section indicates the name the design had during its creation and the second one is a more systematic one which is used today. The first test chips was mostly designed by R. Berner, the second one and the full array chips were a co-designed by the author and R. Berner while most of the DAVIS chips on the wafer were designed by the author (except for 128a, and RGBa).

3.8.1.1 SeeBetter10 / -

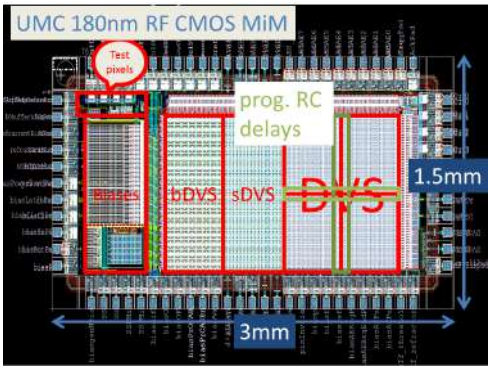


Figure 3.6: Layout of the SeeBetter10/11 showing the arrangement of the different pixels.

The first generation in this series of chips (SeeBetter10) is actually not a DAVIS design but rather a test array of multiple designs used to test certain aspects of the upcoming designs. The chip hosts a heterogeneous array of pixels (as shown in Fig.3.6): 16x32 bDVS pixels (29u pitch), 16x32 sDVS pixels (29u pitch), four different DVS designs arranged in 32x32 pixel quadrants (14.5u pitch): original, 33cas, ls, 33sf with the quadrants separated by programmable delay lines (Tab.3.1). These pixels were used to observe the following design aspects:

- *bDVS:* The *b*ig DVS is a pixel design that contains a log intensity readout on top of the DVS circuitry. This design was used to evaluate the suitability of this intensity readout mode.

- *sDVS:* A *s*ensitive DVS pixel design that is not working most likely because of front end oscillations.

- *original DVS:* A problem that arose when implementing the DVS design in a 180nm fabrication process was that the transistor off current for this process are bigger than the small photocurrents in dark light conditions. Under these conditions the front end stops working because the photoreceptor output voltage node (*pr*) in the front end is clipped

to ground. For this reason multiple alternative designs to cope with this problem were designed while the original was used as reference.

- *ls DVS:* A level-shifting source follower is introduced between the photodiode and amplifier output. This design was unsuitable because it oscillates at low light intensities.

- *33cas DVS:* The front end used 3.3V transistors and a cascode transistor is added between the amplifier and the output to raise the output voltage. This design also leads to low light oscillations because the source follower is avoided and AER reset couples into the photodiode.

- *33sf DVS:* The front end transistors are replaced with 3.3V transistors, the source follower buffer is kept and the cascode transistor omitted. The higher thresholds of the 3.3V transistors allows to operate the pixel also under low light conditions so this pixel variant was chosen for the new designs.

The DVS variants are horizontally and vertically separated by delay cells that allow delaying AER signals through an RC delay circuit of programmable size. These delay cells were included to investigate scaling effects especially how delays in the AER communication are affected by bigger array sizes. The handshake has been shown to work reliably even under long delays.

3.8.1.2 SeeBetter11 / -

Due to a mistake in the fabrication process, there were short circuits in the metal 6 layer of the SeeBetter10 design so it never worked. The same design was re-fabricated with a minor correction in the AER circuits but the first batch of chips were bonded incorrectly. The second batch of SeeBetter11 finally allowed the circuits to be tested.

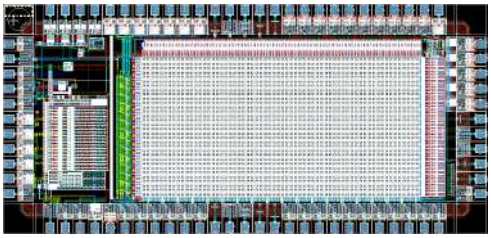


Figure 3.7: Layout of the SeeBetter20 / DAVIS 64a.

Table 3.1 SeeBetter10/11 Specs

Process	UMC MM 1P6M 180nm
Die Size	3mm x 1.5mm
Array size	16x32 / 16x32 / 4x 32x32
Pixel	bDVS / sDVS / DVS Variants
Pitch	29um / 29um / 14.5um
Photodiode	N Well

3.8.1.3 SeeBetter20 / DAVIS 64a

Although the logarithmic readout in the bDVS pixel of SeeBetter10/11 was working and converting light intensities from a illuminance range of 4 decades onto a voltage range of 450mV, the mismatch was too strong even after off-chip calibration. For this reason, it was replaced by a conventional APS readout that allows correlated double sampling which reduces the noise significantly. This first DAVIS pixel initially called APS-DVS (on SeeBetter20/DAVIS64a) was a modified bDVS pixel with 29um pitch so that it could be designed quickly. 18 columns (2nd quarter from left in images) had a slightly different layout in which the cascode transistor between the APS and the DVS circuits is longer (no significant effect). The chip contained 64x32 pixels and two shift registers to read out the APS intensity samples (Tab.3.2).
The SeeBetter20 also hosted the first addressable bias current generator with a coarse-fine current splitting technique (Yang et al. 2012). The previous designs (Delbruck et al. 2010a) used a long shift register chain which was replaced by an addressing scheme that allows increasing the programming speed for single biases and reduc-

ing the layout area. The coarse-fine structure allows to span a dynamic range of over 170dB.

Table 3.2 SeeBetter20/DAVIS 64a Specs

Process	UMC MM 1P6M 180nm
Die Size	3mm x 1.5mm
Array size	64x32
Pixel	DAVIS
Pitch	29um
Photodiode	N Well

3.8.1.4 SBret10 / DAVIS 240a

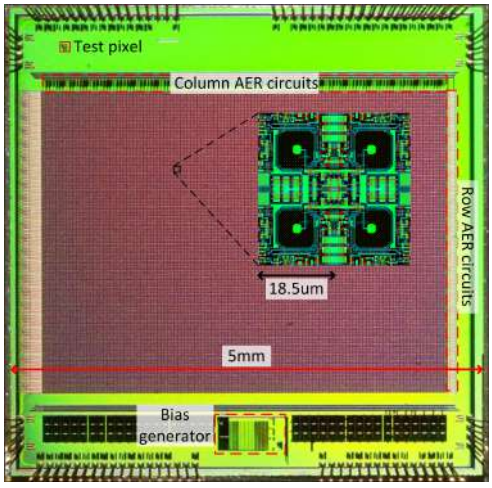


Figure 3.8: Die photo of the DAVIS 240a. The inset shows the layout of a set of 2x2 mirrored pixels. Reprinted from (Berner et al. 2013b)

The success of the DAVIS 64a design lead to the decision to use this pixel design for a bigger array of 240x180. Therefore the pixel size was reduced to a 18.5um pitch and a transfer gate was added to allow for global shutter operation. Due to an error in the on-chip readout logic (the pixels cannot be reset in parallel but only column-wise), the global shutter is not working. This chip as well as the previous chips were fabricated in UMC's 6M1P 180nm standard logic process. This chip is further characterized in (Berner et al. 2013b; Berner et al. 2013a).

Table 3.3 DAVIS 240a Specs

Process	UMC MM 1P6M 180nm
Die Size	5mm x 5mm
Array size	240x180
Pixel	DAVIS
Pitch	18.5um
Photodiode	N Well

3.8.1.5 SBret20 / DAVIS 240b

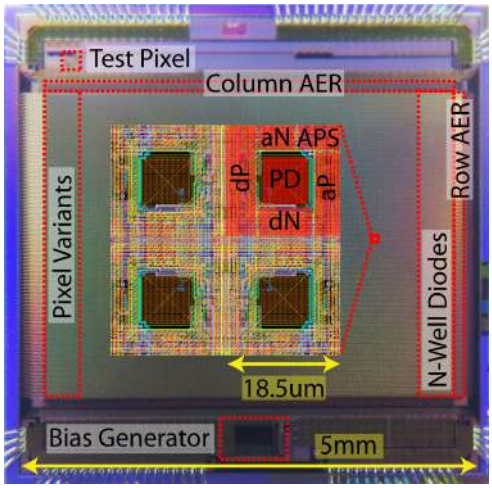


Figure 3.9: Die photo of the DAVIS 240b. The inset shows the layout of a set of 2x2 mirrored pixels. Reprinted from (Brandli et al. 2014a).

The DAVIS 240b design was ported to the TowerJazz 180nm 6M1P CMOS image sensor process to asses its suitability for the upcoming wafer run. To investigate the quality of the optimized surface photodiodes supplied in this process, a subset of 20 columns contained conventional N well photodiodes while the rest was using the dedicated CIS surface diodes. The surface diodes show a higher quantum efficiency which can be seen in APS images (pixel of N well diodes are darker) and in the reduced DVS noise in low light conditions.
Another subset of columns contained 4 different variants of the pixel design to investigate design variations (from left to right on the chip, right to left in the display):

- *20 PosFB:* By adding pull down circuits to the comparators, they become state-holding so that once they exceed a threshold, the request is latched and the full AER handshake has to be completed before request pull-down line can be released. This state-holding pixel avoids events that only request in the row direction, then go below the threshold so that as soon as the row acknowledge arrives, they do not request in the column direction. Since this pixel variant is functional it became the basis for most of the following designs. The schematic for this pixel can be found in the appendix as Fig. A.4.

- *10 More pr gain:* By adding a diode-connected transistor between the feedback transistor and the photodiode, the gain of the photoreceptor is increased. The pixel exhibits a higher contrast sensitivity but it is also more noisy, has a lower dynamic range (bright part of scene) and exhibits spatial mismatch patterns in the event output when exposed to artificially illuminated surfaces. The schematics of this pixel can also be found in the appendix as Fig.A.5.

- *10 Less GS coupling:* The layout of this pixel is slightly modified to include a lateral M4 shield (grounded) between the global shutter signal TX and the photoreceptor output node (*pr*). The according layout modification can be found in the appendix as Fig. Fig.A.6. This change in design does not not significantly reduce the number of TX-induced events because it does not shield the right signal. For the following designs on the SeeBetter wafer, the capacitance that actually led to the coupling was identified and shielded.

- *10 More diff gain:* Instead of a cap ratio of about 1:25, this design variant contains a ratio of about 1:40. The pixel exhibits more contrast sensitivity but not as much as the "More pr gain" pixel. This pixels seems to be a valid alternative to the lower gain design. This pixel contains the same M4 shield as the "Less GS coupling" pixel which can also be seen in the APS readout. The reset signal and the read signal of these two columns have a different voltage than the other columns. This offset is most likely caused by the increased readout capacitance but it is removed by the correlated double sampling and therefore not seen in the final picture.

This chip is further characterized and described in (Brandli et al. 2014a)

Table 3.4 DAVIS 240b Specs. **20 columns are using N well photodiodes.

Process	TowerJazz CIS 1P6M 180nm
Die Size	5mm x 5mm
Array size	240x180
Pixel	DAVIS
Salicide block	No
Pitch	18.5um
Photodiode	Surface**

The DAVIS 240b design served as basis for the DAVIS designs on the SeeBetter wafer which contains designs described in the following subsections. The designs on the wafer are indicated with a * in the title and at the moment this thesis is written they are produced but not yet tested.

3.8.1.6 apsDVS128* / DAVIS 128a

This mini-DAVIS of S. Bamford and D. Moeys with the dimensions 3.5mm x 3.8mm contains an array of 128x128 pixels of the DAVIS 346b type.

Table 3.5 DAVIS 128a Specs

Process	TowerJazz CIS 1P6M 180nm
Die Size	3.5mm x 3.8mm
Array size	128x128
Pixel	DAVIS PosFB
Salicide block	Yes
Pitch	18.5um
Photodiode	Burried, Deep P Well

3.8.1.7 PixelParade* / DAVIS Sense 192a

This chip by D. Moeys and S. Bamford with the dimensions 5mm x 5mm contains multiple variations of DAVIS pixels including more sensitive DAVIS pixels.

Table 3.6 DAVIS Sense 192a Specs

Process	TowerJazz CIS 1P6M 180nm
Die Size	5mm x 5mm
Array size	208x192
Pixel	Various
Salicide block	Yes
Pitch	18.5um
Photodiode	Various

3.8.1.8 SBret21* / DAVIS 240c

The DAVIS 240c is a copy of DAVIS 240b but the test columns have been replaced and the parasitic coupling between event readout and photoreceptor has been removed. At moment of the latest revision of this thesis (Feb 2015) this chip was tested successfully and can be considered functional.

Table 3.7 DAVIS 240c Specs

Process	TowerJazz CIS 1P6M 180nm
Die Size	5mm x 5mm
Array size	240x180
Pixel	DAVIS
Salicide block	No
Pitch	18.5um
Photodiode	Surface

3.8.1.9 apsDVS344* / DAVIS 346a

This 8mm x 6mm chip design contains an array of 346x240 pixels without salicide block and deep P well but in contrast to DAVIS 240c it uses a buried photodiode.

Table 3.8 DAVIS 346a Specs

Process	TowerJazz CIS 1P6M 180nm
Die Size	8mm x 6mm
Array size	346x240
Pixel	DAVIS PosFB
Salicide block	No
Pitch	18.5um
Photodiode	Buried

3.8.1.10 apsDVS344b* / DAVIS 346b

DAVIS 346b is a copy of DAVIS 346a but with a different pixel: it uses salicide block and deep P well.

Table 3.9 DAVIS 346b Specs

Process	TowerJazz CIS 1P6M 180nm
Die Size	8mm x 6mm
Array size	346x240
Pixel	DAVIS PosFB
Salicide block	Yes
Pitch	18.5um
Photodiode	Buried, Deep P Well

3.8.1.11 SBretFinal* / DAVIS BSIa

The DAVIS BSIa is a copy of the DAVIS 346b design except for dedicated pads that allow backside illumination (BSI) processing.

Table 3.10 DAVIS BSIa Specs

Process	TowerJazz CIS 1P6M 180nm
Die Size	8mm x 6mm
Array size	346x240
Pixel	DAVIS PosFB
Salicide block	Yes
Pitch	18.5um
Photodiode	Buried, Deep P Well

3.8.1.12 rgbDVS* / DAVIS RGBa

The heterogeneous array on this chip by C. Li contains an array of 320x240 super pixels: 3x APS pixel underneath a red, green and blue color filter together with a DAVIS pixel without color filter.

Table 3.11 DAVIS BSIa Specs

Process	TowerJazz CIS 1P6M 180nm
Die Size	8mm x 6mm
Array size	320x240 / 640x480
Pixel	DAVIS PosFB / APS
Salicide block	Yes / No
Pitch	20um
Photodiode	Buried, Deep P Well / Pinned

3.8.1.13 apsDVS640* / DAVIS 640a

This VGA (640x480) version of the DAVIS has wider power lines and additional power pads to ensure a uniform power distribution across the array.

Table 3.12 DAVIS 640a Specs

Process	TowerJazz CIS 1P6M 180nm
Die Size	10.5mm x 13.5mm
Array size	640x480
Pixel	DAVIS PosFB
Salicide block	Yes
Pitch	18.5um
Photodiode	Buried, Deep P Well

3.8.2 DAVIS Pixel

The main principle behind the DAVIS pixel is to reuse the photocurrent of the DVS for a light intensity readout with an active pixel sensor (APS) circuit. To avoid that the APS readout affects the operation of the DVS, the two circuits are separated by a cascode transistor (MN5 in Fig.3.10). To save power and design area, the DVS circuits are operated at 1.8V while the APS circuit is operated at 3.3V to provide sufficient headroom. As previously described, the photoreceptor uses 3.3V transistors (thick gates in Fig.3.10) to avoid big leakage currents that affect the operation under low light conditions.

3.8.2.1 DAVIS Pixel Layout

A typical DAVIS pixel layout is shown as inset in Fig.3.8 and Fig.3.9: The pixels are mirrored horizontally as well as vertically and grouped into sets of 2x2. This allows to share wells and group analog and digital parts of the pixels together which saves area and reduces noise.

A very crucial aspect in the layout of a DAVIS pixel is to avoid that the digital signals couple into the analog front end of the pixel:

In the sDVS pixel of SeeBetter10/11, the acknowledge line ran over the photodiode and thereby coupled into the photoreceptor output node (*pr*) which can cause further events. By moving all digital lines away from the photodiode as in the designs following SeeBetter10/11, this problem can be avoided.

Another effect observed in the pixels of SeeBetter10/11, is a feedback coupling from the differentiator output (Vdiff in Fig.3.12) to the photodiode which leads to oscillations that can cause events. Without parasitic capacitance extraction, these effects cannot be observed in simulation which is why DAVIS 240a and following designs were designed in Cadence which allows do post-layout simulation including parasitic capacitances.

In the DAVIS 64a pixel, the APS readout node passed close to the photodiode which is an explanation for the APS readout triggered events in this design.

These coupling issues have been resolved for the pixel layout of DAVIS 240a so that a rolling shutter frame readout does not trigger events. But there is some capacitive coupling between the transfer gate signal TX and the Vpr node so that the global shutter readout in DAVIS 240b triggers events as seen in Fig.3.11. The source of this coupling was identified and removed for the chip designs on the SeeBetter wafer.

3.8.3 DAVIS DVS Circuits

The DVS circuit within the DAVIS pixels is similar to the original one (Lichtsteiner et al. 2008) (Fig.3.4) but a major improvement results from the reset decoupling introduced by R. Berner. While the original DVS released the request only when the pixel was completely reset (i.e. the comparators turned off), the new scheme releases the request immediately upon the arrival of both acknowledges. Fig.3.12 shows the de-

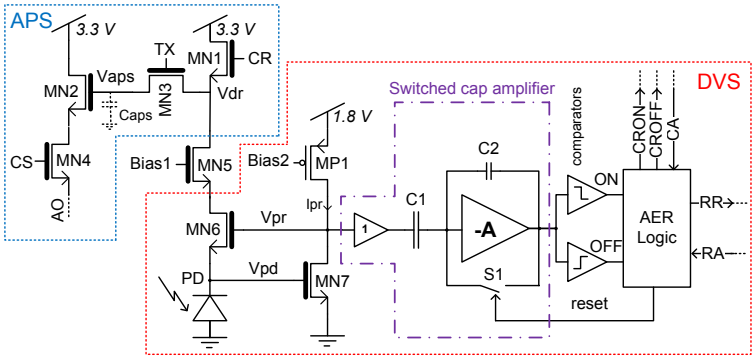


Figure 3.10: Simplified schematics of the DAVIS pixel. Transistor level schematics can be found in Fig.3.12. Reprinted from (Brandli et al. 2014a).

tailed schematics. As soon as both acknowledge signals arrive at the pixel, the nReset capacitance is discharged and the request lines are no more pulled down. This decreases the pixel handshake time and thereby increases the chip AER bandwidth. This development together with the development of the word serial AER scheme (Berner 2011) increased the bandwidth of the AER significantly. The timing in these circuits is highly sophisticated and their development took several years because of their intrinsic complexity. This development was the main reasons for the delay of the next generation functional vision sensor after the DVS.

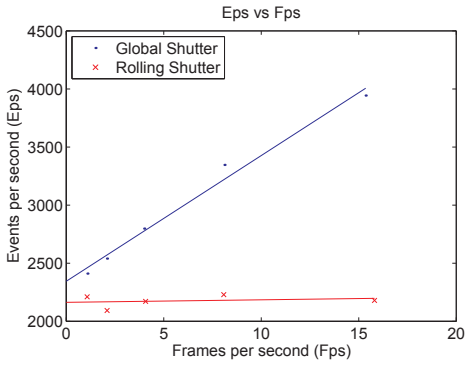


Figure 3.11: Event rate versus APS readout rate in the DAVIS 240b chip. Reprinted from (Brandli et al. 2014a).

3.8.3.1 Characterization

The DVS readout can be characterized using multiple measures. The following characterizations are based on the original DVS measurement protocols (Lichtsteiner et al. 2008).

Latency

In (Posch et al. 2011b) and (Serrano-Gotarredona et al. 2013b) latency was measured as the time between the onset of a blinking LED signal and the first event of the address range covering the LED on the focal plane. This measure is prone to outliers and neglects the mismatch in latency. For the latency measurements of DAVIS 240a/b a different measure was used: the latencies of the first event of all pixels exposed to the blinking LED are averaged and the median value is reported. Fig.3.13 shows how the latency decreases with increasing light intensities and the inset shows the latency distribution. This decrease in latency is because the front end can react faster if the photocurrent is bigger. For DAVIS 240a, a latency of 12us was reported while it is 3us for DAVIS 240b because the biases used for DAVIS 240a were not fully optimized for speed. Extensive testing of DAVIS 240b showed that for latencies below 12us, the comparators become the latency "bottleneck" and

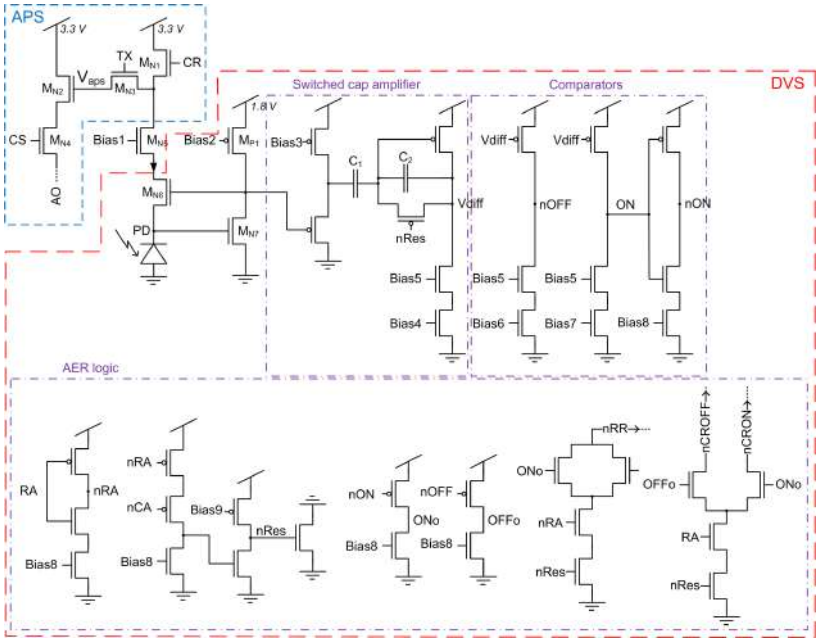


Figure 3.12: Detailed schematics of the DAVIS pixel. For the designs on the SeeBetter wafer, the diff cascodes i.e. Bias5 gates are omitted.

if they are biased hard, the pixel can react even faster.

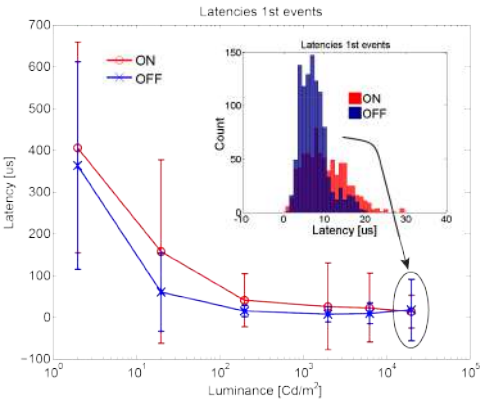


Figure 3.13: Latency measurements of DAVIS 240b. Inset shows the statistical distribution of the first events in the field of view.

Event Contrast Threshold Mismatch

To measure the mismatch in contrast threshold, the array is exposed to a uniformly modulated light source i.e. an LED operated with a slowly-modulated sinusoidal current source and uniformly distributed across the chip using an integrating sphere. By counting the events, the threshold and its variation across pixels can be determined. For DAVIS 64a this mismatch is 3% while it is 3.5% for the DAVIS 240a/b and the difference is most likely due to differences in the measurement protocol.

3.8.4 DAVIS APS Circuits

The new APS readout feature works completely independent of the DVS circuits; they are explained and characterized in the following.

3.8.4.1 Readout

To capture an image, the APS circuit has to be reset, the conversion capacitance with voltage V_{aps} is charged and during the exposure the

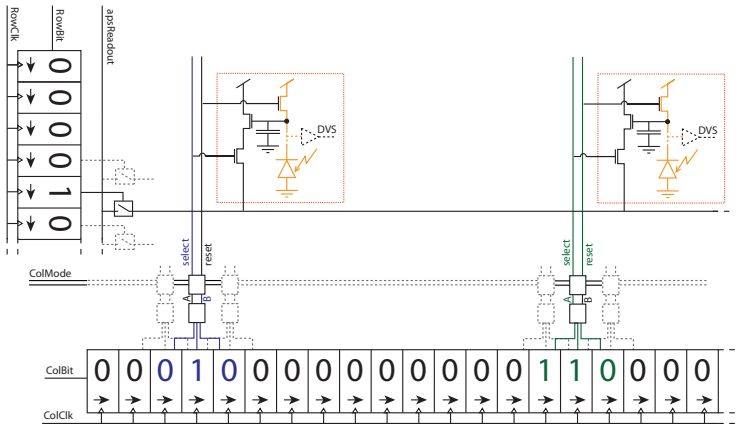


Figure 3.14: APS readout scheme. From (Brandli et al. 2014a)

photocurrent is integrated as seen in Fig.3.15. Due to transistor mismatch and kTC noise, the reset voltage is not constant across pixels and to remove this fixed pattern noise (FPN) a double-sampling scheme is employed: Instead of only measuring the voltage after the exposure, it is also measured after the reset and the difference corresponds to the signal. Stronger light integrates the charge faster and the voltage V_{aps} drops faster while this takes longer for darker pixels.

Overexposed pixels are pixels in which the charge on the conversion capacitance is not sufficient to supply the photocurrent during the full length of the exposure. In these overexposed pixels, the DVS part should still be functional which can be achieved by using an overflow protection. This protection is performed by setting the reset signal CR to a non-zero overflow bias voltage during the exposure: as soon as V_{aps} drops below this voltage to the overflow level, it turns on the reset transistor M_{N1} and it supplies the photocurrent.

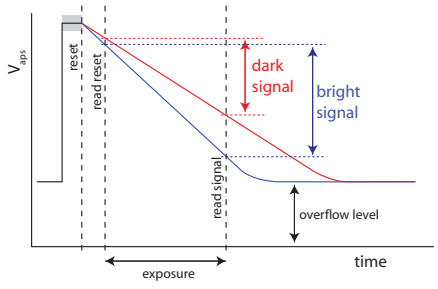


Figure 3.15: APS readout signal. The mismatch leads to an offset in the reset signal (gray shading) which can be removed using a double sampling scheme.

Rolling Shutter Readout

To allow for correlated double sampling that removes kTC noise the first readout scheme implemented in DAVIS 240a was a rolling shutter. The first generations of DAVIS (64a, 240a, 240b) do not contain an on-chip ADC therefore two shift registers are used to select the pixel which is connected to the analog output pad and the off-chip ADC.

Most conventional image sensors readouts select a row at a time and then read the according pixel values either in serial or column-parallel. The APS readout in the DAVIS instead selects a column to scan it out row by row; this is a legacy of previous layouts in which this scheme fitted better onto the chip. The correlated double sampling scheme is done in software by first converting a reset read which is stored in memory and then subtracting the signal read.

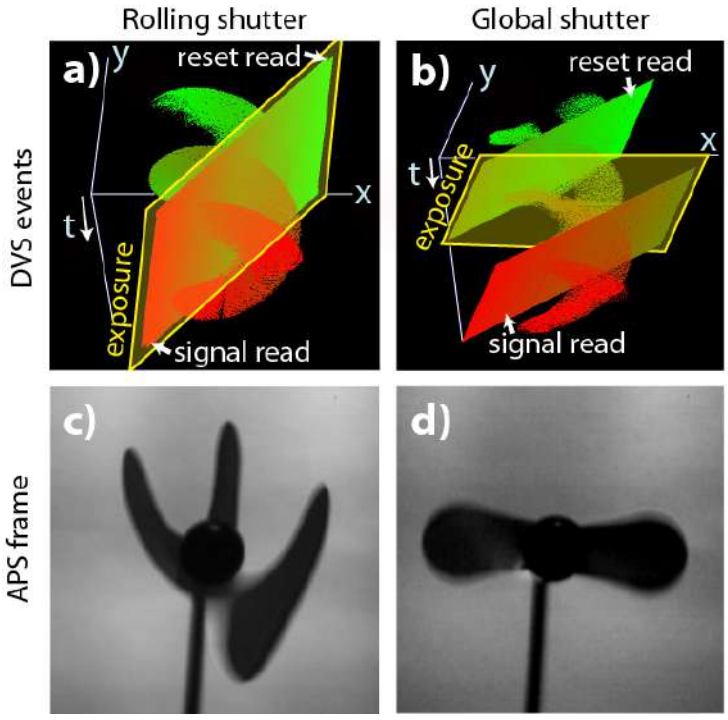


Figure 3.16: Rolling vs. global shutter readout: Rotating fan at 50Hz in event DVS space-time view and APS frames. a) The exposure of the scene is spread in time and therefore the exposure plane is tilted and cuts the DVS trace multiple times which leads to the artifacts seen in c). The global shutter allows to expose all pixels at the same time so b) the exposure plane is not tilted and thereby d) reduces the motion artifacts. Reprinted from (Brandli et al. 2014a).

The pixels routed to the ADC pad are selected using a 2D shift register (as shown in Fig.3.14): A bit pattern in the column shift register determines which column is reset and read, while the row shift register iterates over the rows in the selected column. The exposure in the rolling shutter scheme corresponds to the time between the reset and the signal read. To allow short exposures, the reset read and the signal read are performed on different columns. To distinguish the two columns, bit patterns in the shift register determine the reset column (110) and the signal column (010). The number of columns between these two patterns multiplied with the time it takes to read out a column corresponds to the exposure. To determine whether the reset column should be reset or selected for readout or whether the signal column should be selected for readout, each column contains a logic block which takes so called column state lines (ColState) as input. These signals are shared among all column logic blocks to determine whether the column select signal or column reset signal should be raised:

- **00 : Null.**The Null state does neither select or reset a pixel and is used to avoid glitches between the state transitions. It is inserted between any of the states.

- **01 : Reset Read.**The column with the bit pattern 110 is selected and connected to the source follower transistors at the end of the row.

- **10 : Signal Read.**The column with the bit pattern 010 is selected to be hooked up to the source followers.
- **11 : Reset.**The column with the bit pattern 110 is reset by shorting the readout capacitance to Vdd.

The rolling shutter column sequence for the ColState is the following: Null - Reset - Null - Reset Read - Null - Signal Read - Null - Advance a column (clock column shift register). Reset Read and Signal Read are the states in which a full column is read out by clocking a bit through the row shift register: either the reset column (column bit pattern 010) or the signal column (column bit pattern 110). The detailed state machine diagram can be found in the appendix.

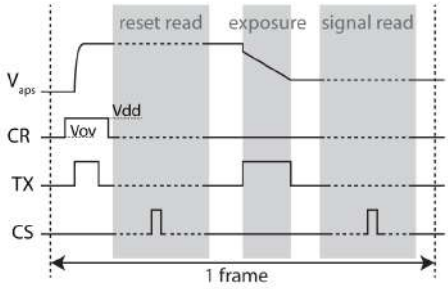


Figure 3.17: Global Shutter Timing Diagram. Reprinted from (Brandli et al. 2014a).

Global Shutter Readout
In contrast to the rolling shutter readout, the global shutter readout separates the reset read in time from the signal read (shown in Fig.3.17) and only one bit pattern (010) is required for the column selection. First the whole array is reset using CR and the reset voltage is sampled using the TX signal, then the full reset frame is read out. The TX gate is connected and the sampled charge is integrated during the exposure and sampled again by disconnecting TX. The signal frame is read out and the reset frame is subtracted in software. The detailed readout state machine can be found in the appendix.
While the global shutter readout shows much weaker motion artifacts as shown in Fig.3.16, it does not correct for kTC noise. The reset signal is sampled in the first TX sampling pulse which introduces kTC noise but this noise is not read out twice which would be necessary to cancel it in a correlated double sampling scheme. Instead, the second TX pulse used to sample the signal, introduces new kTC noise which is uncorrelated to the one of the reset sample.

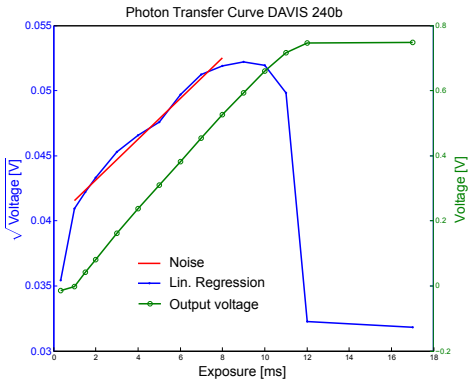


Figure 3.18: The photon transfer curve of the DAVIS 240b.

3.8.4.2 Characterization

Photon Transfer Curve (PTC) & Conversion Gain
To characterize the response of an APS imager, the photon transfer curve (PTC, Fig.3.18) is a powerful tool: for increasing exposures, the frames of the imager are captured and analyzed. The imager is facing a uniform light source (integrating sphere) which allows to compare the responses of different pixels and thereby characterize the different noise components as well as other key figures. Since the shot noise grows quadratically with the signal, it can be used to infer the conversion gain (Janesick 2007):

$$s_{tot}^2 = k^2 s_R^2 + k(S_{tot} - S_{off}) \qquad (3.2)$$

where s_{tot} is the total noise (V), k the system conversion gain (V/e^-) or ADC sensitivity (slope of the noise fit in Fig.3.18), S_R the temporal noise of the readout channel (e^-), S_{tot} the output signal (V) and S_{off} the signal offset (V).

The system conversion gain of SBret240b is $8.8uV/e^-$ which corresponds to a conversion capacitance of 18fF which matches the capacitance extractions from post layout simulations.

Dynamic Range (DR)
The dynamic range is defined as the maximum signal level divided by the noise floor i.e. the noise at minimal signal amplification:

$$DR = 20 \cdot log\frac{S_{max}}{s_{min}} \tag{3.3}$$

With optimized biases for APS output swing (DVS still functional), S_{max} reaches 1563mV and the temporal noise at minimal exposure s_{min} is 1.3mV which leads to a dynamic range of **61.3dB**. The actual dynamic range is limited by the quantization noise of the ADC and since only about 8.5bits of the ADC are actually used, it just reaches 51dB.

Signal-to-Noise Ratio (SNR)
While the dynamic range compares the maximal signal with the smallest detectable signal (intra-scene dynamic range), the signal-to-noise ratio compares the ratio of signal versus noise at a specific signal (usually at 50% of S_{max} but it can also be reported at the saturation as in (Brandli et al. 2014a)):

$$SNR = 20 \cdot log\frac{S}{s} \tag{3.4}$$

The highest SNR in DAVIS 240b is achieved at the saturation of 600mV (with the biases used in (Brandli et al. 2014a)) where the noise level is $320e^-$ leading to a SNR of 46dB. The SNR is smaller that the DR because the shot noise increases with the signal intensity.

FPN
The fixed pattern noise is also measured using the PTC by computing the standard deviation of the uniformly illuminated pixels within an averaged frame; this averaged frame is computed across many frames to remove the temporal signal and noise components. For the DAVIS 240b this value reached a maximum just below saturation with a value of 0.5%. This value also includes the photo response non-uniformity.

3.8.5 DAVIS Camera

To test the DAVIS chips as well as for algorithm and application development, the chip was integrated into a camera consisting of a PCB, a Lattice MachXO CPLD, a Cypress FX2 USB 2.0 communication chip, power regulators, a lens mount, a lens, a Texas Instruments THS1030 30 Msps 10 bit ADC, multiple probe access points and the packaged chip in a socket (Fig.3.20a). After the chip was working, the design of this test camera was simplified and redesigned by inilabs to be more compact (Fig.3.20b) and to integrate an inertial measurement unit (IMU) (Delbruck et al. 2014).

3.8.5.1 CPLD Logic

The CPLD is the direct interface to the DAVIS chip and used to configure it and to transmit the events to the USB chip.

Timestamping
To use the asynchronous events in conventional computers, they not only have to be synchronized but they also require a timestamp so that they can be communicated in packets and logged or processed offline. The synchronization and timestamping is performed by the CPLD which contains an internal clock with microsecond resolution and on the arrival of an event its address is registered as a 16bit word and the arrival timestamp is stored in another 16bit word (as shown in Fig.3.19). In case of a timestamp, the first two bits (MSBs) are set to 01 and the following 14bit encode the actual timestamp but since 14bit would only cover 16ms, every time the counter is reset, a so called wrap event (MSBs = 10) is sent. These wrap events allow to extend the timestamp to 32bit in software.

ADC Samples
The simplest way to communicate and debug the grayscale values of the APS image is to communicate the ADC samples in the same way DVS events are communicated but instead of an address, the ADC output is sent. This scheme was implemented first but has the drawback that the pixel address of the ADC sample is lost and that

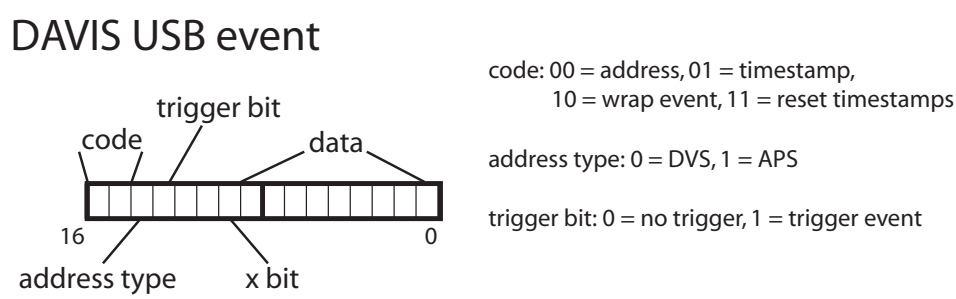


Figure 3.19: The DAVIS data format of the 16bit words sent through the USB connection.

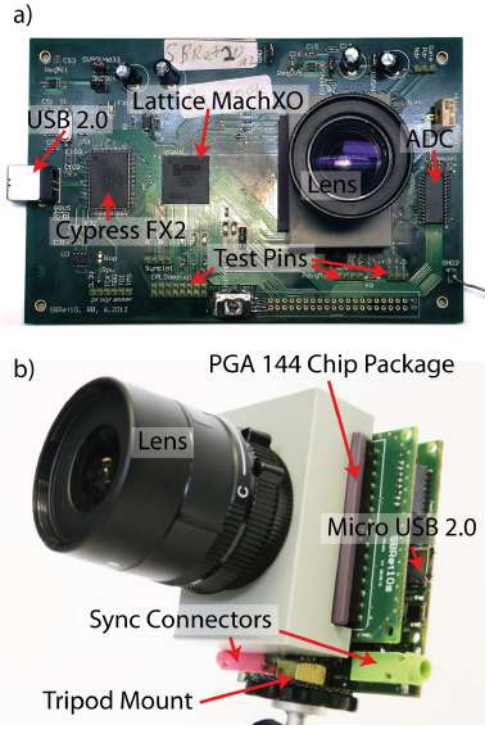


Figure 3.20: Picture of the DAVIS cameras: a) Testboard developed by R. Berner. b) More compact version of the board developed by inilabs including IMU. Adapted from (Brandli et al. 2014a).

a 10bit value requires 32bit (16 bit ADC word + 16bit timestamp word) to be communicated. The resulting high data volume make it probable that the buffers overrun and samples are lost which leads to a misalignment of the pixel values and the full image becomes corrupted.

3.8.5.2 Software

All recordings and most algorithm development for the DAVIS have been done in the Java-based software framework jAER (*jAER*). Even though the software was already fully functional for the DVS when the first DAVIS were tested, many features had to be added or modified to handle the DAVIS data and the complex configuration of the board including the APS readout.
A DAVIS chip class had to be implemented to acquire and process the events. To configure the camera, the configuration class was modularized to reuse code of previous cameras and adapted to the new camera. An extensive graphical user interface (GUI) had to be built to allow the configuration of all DAVIS parameters. A so-called hardware interface class had to be created so translate the DAVIS USB events into DAVIS raw events which carry a 32bit timestamp and either an address or an ADC sample (shown in Fig.3.21). These raw events can be logged into files or streamed over a TCP or UDP port. Methods to extract and handle the DVS events as well as the APS frames in Matlab were implemented. The "ApsDvsFrameExtractor" class in jAER gets the frames out of the DAVIS data stream and a javacv interface allows to run

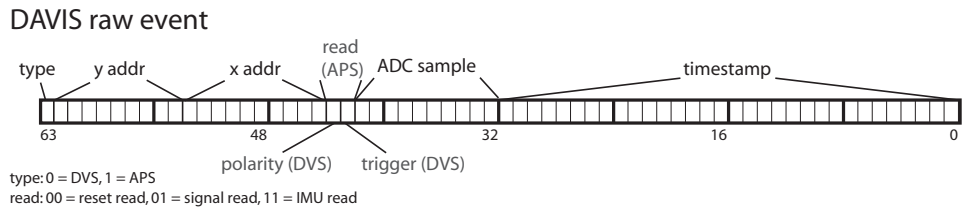


Figure 3.21: The DAVIS data format of the 2x32bit words used to log events in the jAER software.

methods from the OpenCV computer vision library on them (Bradski 2000). To allow using the event filters developed for the DVS also for the DAVIS data, the default event iterator had to be modified to skip APS sample events without losing them from the packet.

For more efficient rendering that buffers the events independently from the APS image frames, the existing pixel-based approach was replaced. Instead of drawing the float buffers using the glDrawPixels method, the events are treated as textures which can be stored in the graphics card memory and displayed by drawing textured quads on top of each other (as shown in Fig. 3.22).

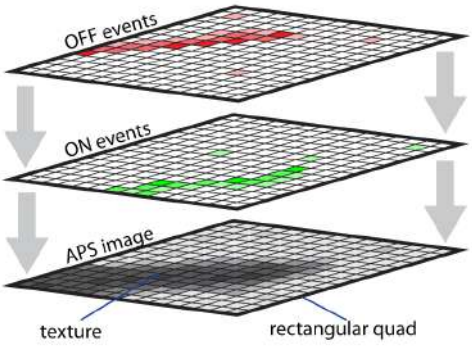


Figure 3.22: Texture-based rendering of the events as a stack of quads.

3.8.6 Results

The first iterations of the DAVIS designs were tested and characterized to identify faults and design errors. The key findings and exemplary data of the first three DAVIS generations is presented in the following.

SeeBetter10

Even though not an actual DAVIS chip, the key findings of R. Berner on the SeeBetter10 test chip were crucial for the development of the DAVIS: The logarithmic readout does not produce acceptable results and by using 3.3V transistors in the phoreceptor, the headroom problem can be solved.

The logarithmic intensity readout of the bDVS pixels was based on a readout scheme developed in (Berner 2011) for dichromatic color pixels. But even though the chip was calibrated in software to account for offset and gain mismatch, the quality of the acquired images was not satisfactory. This result lead to the decision that the upcoming chips should contain a conventional APS intensity readout.

From the data of the first DVS chips in the 180nm technology it became evident that the DVS pixels were slower and had less low-light capabilities (less headroom for the *pr* node) than the 350nm chips such as DVS128. This was traced back to the fact that the lower threshold voltages of the 180nm transistors have more sub-threshold leakage current (off current). If the photocurrent is smaller than the off current of the feedback transistor (MN6 in Fig.3.10), the voltage difference between source *pd* and gate *pr* has to be so low that *pr* is pushed towards ground for small photocurrents. The solution that solved this problem the best is to use 3.3V transistors in the photoreceptor (MN6 and MN7) because the higher threshold voltages lead to less leakage and a higher offset of *pr*.

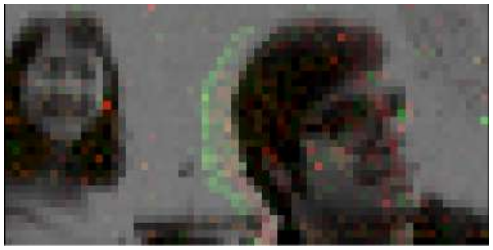


Figure 3.23: DAVIS 64a looking at two heads. Green: ON events, red: OFF events, grayscale: APS frame

DAVIS 64a

Already the first DAVIS iteration produced good results (Fig.3.23) which lead to the decision to use the DAVIS pixel for the upcoming runs of larger pixel arrays. But the data showed a clear coupling of events to the frame readout and by using post layout extraction the according coupling capacitance was identified: the column select line was running over the photoreceptor node *pr*. The pixels variations with a longer cascode transistor between APS and DVS circuits did not affect the output so the shorter gate was used for the following designs.

The measurements on the DVS sensitivity revealed that the temporal contrast sensitivity is with down to 10% better than in the DVS128 at a comparable threshold matching of 3% for ON and 2% for OFF events. This might be an effect of the increased event bandwidth that allows to complete the AER handshake faster.

DAVIS 240a

The second iteration of the DAVIS sensor, the DAVIS 240a (SBret10) with 240x180 pixels can capture wider scenes with a better resolution (Fig.3.25). Unfortunately it is missing a global APS reset signal, so it is not possible to run it in global shutter mode even though it contains the TX transfer gate in the pixel and the according global shutter readout circuitry in the periphery. The chip contained a circuit designed by R. Berner to suppress requests from so-called "hot pixels" that cannot be reset below the threshold and therefore continuously generate events. The hot pixel suppression circuit could not tackle

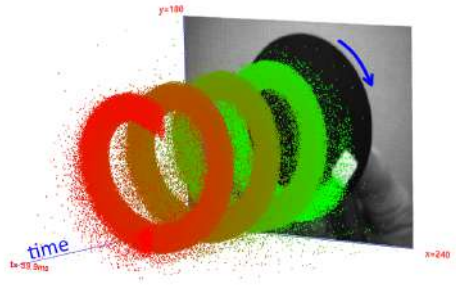


Figure 3.24: 40ms slice of DVS events from DAVIS 240a looking at a rotating disk with 100Hz revolutions. Green: old events, red: new events, grayscale: APS frame. The white dot is captured from a still image of the dot which otherwise would have been blurred. Reprinted from (Berner et al. 2013b).

the hot pixel problem sufficiently well which might according to R. Berner be because these events are generated by charge injection when the pixel reset is released. Fig.3.24 shows how

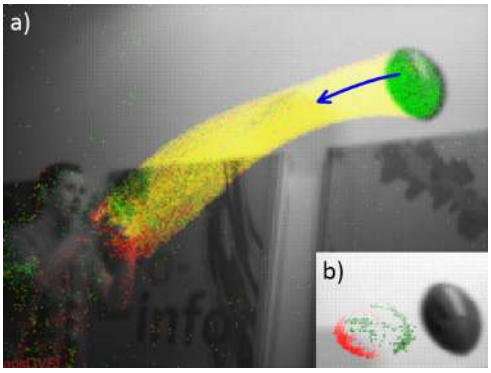


Figure 3.25: DAVIS 240a looking at the author catching a ball. Green: ON events, red: OFF events, grayscale: APS frame. a) Ball leaving an event trace (yellow: overlay of ON and OFF events). b) 5ms slice of DVS events taken 75ms after the APS exposure. Reprinted from (Berner et al. 2013b).

the events continuously follow a moving stimulus even at high speeds. The according trace of the events is nicely illustrated as a space-time 3D

plot. The minimal latency measured for DAVIS 240a was peaking at 12us and a contrast sensitivity of about 12% with about 3.5% matching for ON and OFF events was achieved (Berner et al. 2013b).

The high dynamic range of the DVS is illustrated by capturing a scene under extreme light conditions such as in Fig.3.26. While the conventional DSLR camera and the DAVIS APS readout saturate and can not capture the full scene, the high dynamic range of the events sees contrast under bright and dark light conditions.

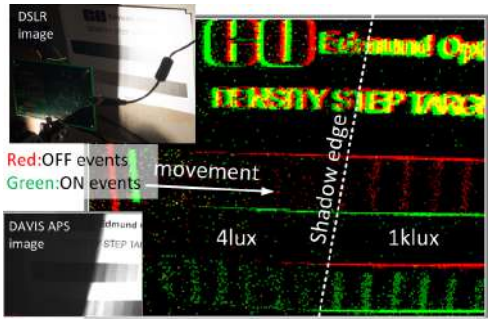


Figure 3.26: 60ms slice of DVS events from DAVIS 240a looking at a moving Edmund density step chart. Green: ON events, red: OFF events. Insets showing picture taken by DSLR camera and the DAVIS APS readout. Reprinted from (Berner et al. 2013b).

DAVIS 240b

Fig.3.27 shows how the APS output of the DAVIS 240b allows analyzing a scene while the DVS output allows tracking motion within such a scene. The key result from DAVIS 240b was that the design is functional in a different process which even improves the performance of the chip. The dynamic range is 130dB which is 10dB better than DAVIS 240a due to the increased quantum efficiency of the surface photodiodes that allows to operate the sensor under darker conditions. The minimal latency is also lower but this is due to a harder biasing of the comparators in the according measurements. Another key finding is that the event-latching positive feedback pixel is functional and it is therefore used for most of the DAVIS designs on the SeeBetter wafer. A detailed overview of the figures of DAVIS 204b can be found in Fig.3.28.

3.8.6.1 Characterization

Power Consumption

The power consumption of the DAVIS 240b chip (without ADC) is between 5mW and 14mW whereas most power is consumed by the digital output pads which are powered by the 3.3V supply: 1.2mW up to 8.3mW for high event activity. Also the 3.3V analog power supply consumes a considerable amount of power when the APS readout is activated: 3.3mW but only 0.1mW with low event activity and APS readout turned off. The digital 1.8V power supply consumes between 0.1mW to 0.54mW. The analog 1.8V power supply consumes 1.17mW up to 1.98mW for high event activity.

Bandwidth

The maximal chip bandwidth (limited by AER) can be estimated by self-acknowledging the chip i.e. hooking up the request to the acknowledge line. The request frequency under bias settings that saturate the DAVIS 240b AER bus, reaches 55MHz but since the chip is using a word-serial protocol, the frequency of the y-address request has to be subtracted leading to a chip bandwidth of about 50Meps (million events per seconds). The maximum of events transmitted through USB is limited by the USB bandwidth and the speed of the CPLD so that the system bandwidth is limited to 12Meps.

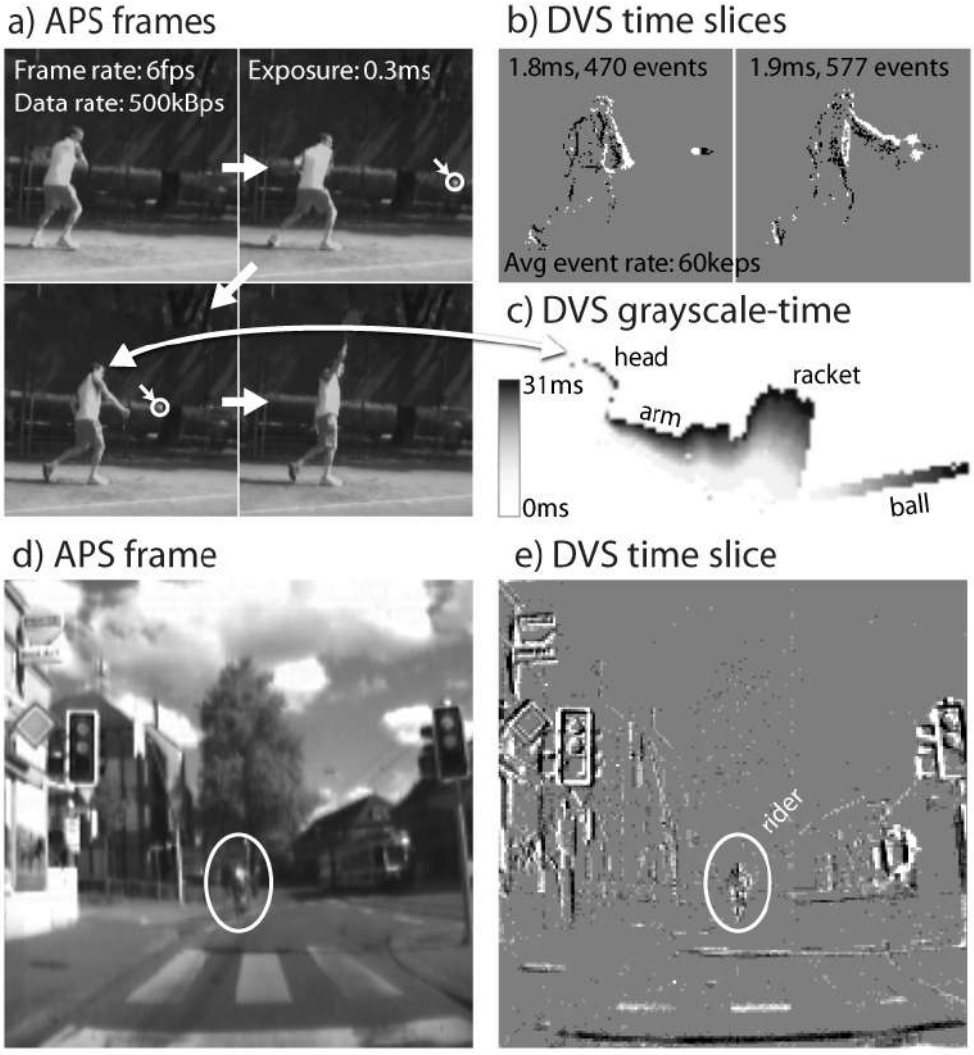


Figure 3.27: Output of the DAVIS240b: a)-c) Tennis player hitting a ball and d)-e) street scene recorded from car dashboard. a) Sequence of APS frames with the ball encircled. b) Temporal slices of accumulated events rendered in black (OFF) and white (ON). c) Zoom in on a 31ms slice of events: the darker, the newer the event. d) APS output and e) concurrent DVS output (134us) showing how it enhances moving objects. Reprinted from (Brandli et al. 2014a).

3.8.7 Discussion and Outlook

The DAVIS is highly suited for machine vision tasks because it fuses global shutter frames for scene analysis with asynchronous temporal contrast events for motion analysis and tracking. It fuses two powerful technologies on the pixel level by reusing the photocurrent of a single photodiode. In comparison to the other technology that combines DVS and intensity readout (ATIS), the DAVIS is better suited for fast motions while the ATIS is suited for surveillance scenarios. This is mainly because the ATIS intensity readout has a high dynamic range but produces severe motion artifacts. The 62% smaller pixel of the DAVIS allows to embed the sensor on mobile platforms, the smaller power consumption ensures a longer battery life and most importantly, the global shutter images allow a detailed scene analysis. In combination with the IMU, the DAVIS camera is the perfect machine vision system because it allows to measure spatial intensities, temporal contrast and ego-motion. With this information it can not only recognize objects but it can also track their movements as well as its own which allows to fully answer the WHAT and WHERE of visual scene understanding on a low power budget and with a short reaction time.
In the future following issues should be tackled:

- The mechanisms that lead to hot pixels should be investigated further and solutions should be elaborated.
- The leakage in the reset transistor that causes background noise events should be removed. Bamford developed a strategy on a design on the PixelParade chip that might be a possible solution to the problem.
- A design routine or guideline to avoid critical coupling between APS and DVS readout should be elaborated.
- The low light performance of the photoreceptor should be improved.

The next chapter gives and overview of how the data of these sensors can be used in algorithms.

	This work	Posch et al. [4]	Serrano-Gotarredona et al.[2]
Functionality	async. temporal contrast + APS	async. temporal contrast + level crossing intensity	async. temporal contrast
CMOS Technology	0.18um 1P6M MIM CIS	0.18um 1P6M MIM	0.35um 2P4M
Chip size mm^2	5 x 5	9.9 x 8.2	4.9 x 4.9
Array size	240 x 180	304 x 240	128 x 128
Pixel size um^2	18.5 x 18.5	30 x 30	30 x 31
Fill factor	22%	20%,10%	10.5%
Pixel complexity	47 transistors, 2 MIM caps, 1 MOS cap, 1 photodiode	77 transistors, 3 caps, 2 photodiodes	N.A.
Supply voltage	1.8V / 3.3V (pixels)	3.3V analog, 1.8V digital	3.3V
Power consumption high activity low activity	14mW 5mW	175mW 50mw	4mW (100keps)
Dynamic range	130dB DVS (0.01 lx), 51dB APS	Intensity 125dB, DVS N.A.	120dB (60dB intrascene)
Min. contrast sensitivity	11%	13% @ 100lx, 30% @1klux	1.5%
FPN	0.5 % APS (1), DVS 3.5% contrast mismatch (2)	<0.25% intensity, DVS N.A.	0.9% DVS
Max. bandwidth	50Meps (self -ack) 12Meps(CPLD), 50 fps	N.A.	20Meps
Min. latency	3us @ 1klux (3)	<4us @ 1klux (fastest pixel)	3.20us @ 2klux (fastest pixel)
APS dark signal	1200e^-/s (4)	N.A.	N.A.
APS readout noise in dark	200e^- (5)	N.A.	N.A.
Notes : (1) Measured by PTC method. (2) Measured using global 3Hz sinusoidal intensity variation from integrating sphere with known contrast. (3) Average latency of the first event of 25 pixels in a small step-pulse-stimulated patch with 1klux focal plane illumination using biases optimized for shortest latency. Latency at 2lux was 400us, and was almost constant above 100lux. (4) Measured by APS dark droop rate and inferred Caps from PTC. (5) Inferred from PTC conversion gain measurement and measured (reset-signal) noise.			

Figure 3.28: Specifications of the DAVIS 240b. Reprinted from (Brandli et al. 2014a).

4 Event-Based Machine Vision Algorithms

Already shortly after the development and release of the first event-based vision sensors, they were applied to real world problems and first algorithms were developed. This chapter gives an overview over these algorithms and also documents some algorithms in detail which were developed during this thesis.

4.1 Machine Vision

Vision plays a crucial role in how we perceive the world: It allows us to gather very detailed information on our environment over long distances. Artificial vision therefore has a great potential in systems processing information on the real world. This section contains an overview over some of the key challenges in machine vision as well as an overview over some of the most successful approaches.

4.1.1 Artificial Vision

There are several fields that aim to implement artificial visions and the boundaries in between them are very unclear. In the following, these descriptions will be used:

- **Artificial Vision:** Set of all man-crafted systems that use visual data (measurements from multiple light-sensitive sensors) to extract information and interpret it. Almost all of these systems are actually computer vision systems.
- **Computer Vision:** Field that aims to extract information from visual data using computers. Most of the work in this field is focused on the development of algorithms and benchmarks. Time and power consumption or applicability are not constraints in computer vision and the main target is to solve computational problems at the highest accuracy no matter the resources they require.
- **Machine Vision:** Field that aims to implement physical systems that use visual information to interact with the world. This field can be considered a sub-field of computer vision because it often employs its algorithms. Machine vision has to handle the real world, so it includes aspects such as geometry, optics, electronics and physics. And while cost, latency and power consumption are not relevant to computer vision, they are crucial to machine vision. A lot of machine vision today is used in production lines and factories for quality control and assembly. The machine vision notion used in the following is wider than just automatic inspection or industrial robot guidance and covers all implementations of computer vision systems that interact in real-time with the world or a user. This notion is based on (Batchelor 1999) but since the field of real-world computer vision applications i.e. machine vision grew out of the industrial sector and now also includes smart phones, gaming platform, cars, robots or wearables.
- **Robotic Vision:** Field that aims to use artificial vision systems to control and guide robots. This field can be considered to be a sub-field of machine vision because it is also strongly defined by its physical constraints.

Event-based processing can improve the efficiency of a processing system and since efficiency is a minor concern in traditional computer vision but a major one in machine vision, this thesis focuses on event-based machine vision. To speak of event-based computer vision can be considered pointless because computer vision focuses on the development of algorithms without considering their implementation and "`event-based'" is an implementation attribute.

4.1.2 Depth Estimation and Correspondence Problem

Depth is a valuable cue in many machine vision applications because it facilitates scene segmentation, tracking and recognition by adding another

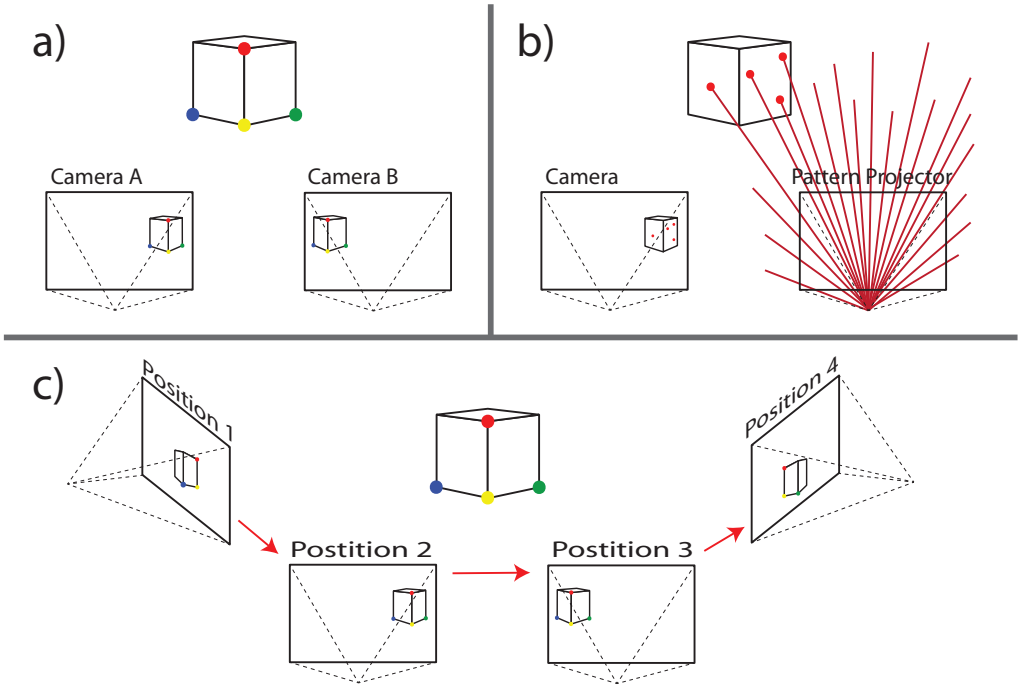


Figure 4.1: Different approaches to solve the correspondence problem for depth estimation. a) Stereovision: Visual landmarks are matched according to their properties (indicated by colored dots) b) Structured Light: A known pattern is projected and observed by a camera c) The camera is moved around the object and the different positions in time allow to perform stereo vision as well as inferring the camera motion.

dimension to the visual data. Most visual scenes are illuminated by an external light source and the only information that comes with reflected photons are its wavelengths and inclination angles. Knowing color and angular position of an object is not sufficient to infer the distance between the object and the camera.

Lidar and Time-of-Flight (ToF) Imaging

One way to access this distance information is to illuminate the scene with a well-timed light source and measure the time the photons travel from source to surface to detector (speed of light is known). This light source has to be very powerful so that its reflections can be detected over long distances and under a strong background illumination. Lidars (portmanteau of light and radar) use a laser as light source and scan the depth information in a scene point by point in one or two dimensions using optical elements such as oscillating mirrors for scanning the scene (Fujii et al. 2005).

Such scanning approaches allow to capture only one distance measurement at a time but if the laser is replaced with a flash that illuminates the full scene at once, time-measuring pixels in so-called time-of-flight (ToF) cameras allow to capture a full depth map in a single shot (Foix et al. 2011). But lidar as well as ToF require light sources which consume a lot of power (20W and more). Another possibility to acquire depth information is integrate geometric constraints i.e. by observing the same scene under different inclination angles.

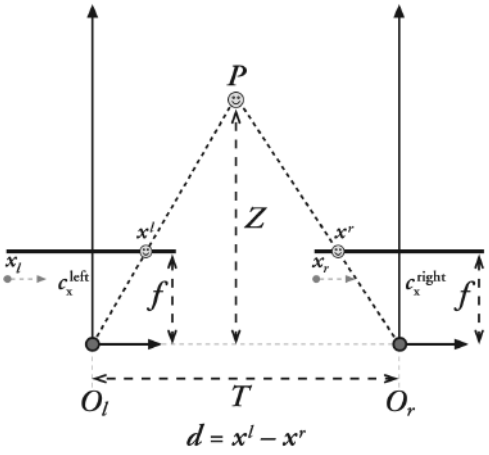


Figure 4.2: Depth estimation in a perfectly undistorted and aligned stereo rig. Reprinted from (Bradski et al. 2008).

Stereo Vision

By knowing the relative angle of a point from two perspectives its distance can be inferred using geometry. This principle is used in stereo vision (Fig. 4.1a): if a point is detected in two cameras, the relative positioning of the two cameras is known so that the inclination angle in the two cameras can be used to triangulate the depth of this point.

Fig.4.2 shows a simplified geometrical drawing how the depth of a certain visual feature (smiley) can be estimated in a perfectly aligned and undistorted setup. The depth Z can be computed if the distance between the imagers T and their focal length f is known using similar triangles:

$$\frac{T-(x^l-x^r)}{Z-f} = \frac{T}{Z} \Rightarrow Z = \frac{fT}{x^l-x^r} \quad (4.1)$$

In Fig.4.2 P is the point of interest, x_l and x_r the x-axis of the left and right imager, x^l and x^r the projection of P on the according image planes, c_x^{left} and c_x^{right} the imager centers, O_l and O_r the imager origins i.e. the focal points and d the disparity. More complicated geometry also allows to account for misalignment and distorted images but the underlying assumption in stereo vision is that the disparity $d = x^l - x^r$ of the points of interest is known.

Correspondence Problem

Getting the disparity for a set of points, means that the these points are captured, identified and matched across both imagers (in at least one image per imager). This is known as correspondence problem and requires to establish point descriptors which allow to compare points between two images. A central problem for establishing such correspondences is that on uniform surfaces it is hard to match pixels or points because they and their neighbors all "look" the same. One approach to solve this problem is to replace one of the two cameras with a light source that projects a known pattern onto the scene which is then observed with the remaining camera.

Structured Light Depth Estimation

If a light source projects a known pattern, the camera output only has to be searched for given projection pattern to compute the disparity between projection and detection. This structured light approach depicted in Fig. 4.1b can also be performed with infrared light to avoid user distraction. Apart from simplifying the correspondence problem, the projected pattern can also be used for depth from focus. Using a projection lens that has different focal lengths for x and y direction creates ellipses whose orientation is dependent on depth (MacCormick 2011).

Structure From Motion

If a camera is moving within a static scene, it is possible to perform stereo vision across the different positions in time as shown in Fig.4.1c. If sufficient points are identified and matched, the system of linear equations becomes overdetermined and it is possible to infer the camera motion as well as the depth of all matched points (Nistér 2005).

4.1.3 Keypoint Detectors and Descriptors

One of the most successful approaches to machine vision that not only solves the correspondence problem and reduces the computational load of tasks is based on keypoint detectors and

descriptors. The idea behind this approach is that instead of working with pixel intensities, characteristic visual point landmarks in an image are identified and described so that they can be matched across images. For this approach to work successfully there are certain requirements for the keypoint detection and description algorithms. Keypoint detection algorithms should ideally identify the same visual landmark in a multitude of images independent of translation, rotation, scale, lens, brightness or color. The keypoint descriptor algorithm also should produce the same description vector for all of these image transformations. It is important that the detector finds characteristic landmarks which are very specific for the observed object so that unnecessary false positive matches can be avoided. For the descriptor it is desirable if the description vectors can easily be matched using a distance metric that can be computed quickly. They should therefore compress the information about the keypoint into a few vector entries.

4.1.3.1 Blob Detectors and Descriptors

The breakthrough of keypoint-based machine vision was initiated by the development of the scale-invariant feature transform (SIFT) by Lowe (Lowe 1999; Lowe 2004). For a stable, rotation-invariant keypoint detection, the SIFT algorithm performs a so-called blob detection: it looks for points in the image with a lot of spatial contrast. Since spatial contrast is independent of the brightness offset, it is invariant to brightness and translation. The Laplacian of Gaussian (LoG) function required for the blob detection is rotation invariant and if it is applied on multiple spatially sub-sampled versions (octaves) of the image; the max of the LoG over the octaves is scale invariant.

To accelerate the LoG, it is approximated with a difference of Gaussian function (DoG). This spatial derivative function is applied with multiple scales and on several octaves of the image. Among these scales and octaves the maxima and minima of the signed DoG are chosen as keypoint candidates. The extreme points than have to be assessed and a first contrast threshold sorts out all keypoints of weak blobs (i.e. maxima/minima with a low DoG). In a second step all maxima from edges are filtered out with a method similar to the one used in the Harris corner detector (Harris et al. 1988): A 2x2 Hessian matrix is used to compute the ratio of the eigenvalues and if they are not similar enough, the keypoint must be on an edge and is sorted out. The remaining keypoints are then used for keypoint description.

The first step of the keypoint description is to evaluate the orientation of the keypoint and to align the description vector to make it rotation invariant. For this the orientation and the magnitude of the spatial derivatives in the neighborhood of the keypoint are calculated. The magnitudes are convolved with a Gaussian centered at the keypoint and added to a histogram with 36 orientation bins. The maximally scoring bin is considered to be the orientation of the keypoint. To get the keypoint description vector (the keypoint's fingerprint), the neighborhood of the pixel is divided into 16x16 sub-regions aligned with the orientation of the keypoint which requires pixel value interpolation for angles that are not multiples of 90°. Each sub-region is assigned to one of the 4x4 region windows which are used to compute orientation histograms with 8 entries (per angular range all contrast magnitudes are accumulated). The resulting 4x4x8=256 orientation entries are normalized by the biggest histogram entry and stored in a vector. For robustness towards brightness, the entries are thresholded and normalized again. A nice tutorial for the better understanding of the algorithm can be found under (*SIFT: Introduction - AI Shack*).

These keypoints and their description vectors can be used for all sorts of machine vision because they allow to solve the correspondence problem by measuring the distance between vectors using vector distance metrics such as L2 or Hellinger metric (Arandjelovic et al. 2012) and matching similar enough keypoints. The success of SIFT lead to further research and improved versions of keypoint detectors and descriptors. One of the most successful approaches because of its speed and robustness is the so called Speeded-Up Robust Features

(SURF, (Bay et al. 2008)) which computes the blob detector and keypoint descriptor using Haar wavelets which can efficiently be computed using integral images.

4.1.3.2 Binary Keypoint Detectors and Descriptors

Even though SURF already sped up the performance of SIFT, embedded real-time applications such as drones demanded even lower computational costs. The approach developed by Rosten delivers this improvement in efficiency by reducing the keypoint detection do a set of binary comparisons. The "features from accelerated segment test" (FAST) corner detector contains a set of binary comparisons to determine whether a pixel is corner(Rosten et al. 2005).

The FAST corner detector is much faster than convolution based approaches because a corner is defined as a set of binary comparisons. This is illustrated in Fig.4.3: The pixel C is a corner if at least 12 contiguous pixels on the ring around C (labeled 1-16 in Fig.4.3) are either brighter or darker than C. This can be computed very quickly using a binary decision tree for each pixel which compares pixel intensities in an optimized order. This corner detector is also used by our collaborators at the Robotics and Perception Group of the University of Zurich (Forster et al. 2014). But not only binary keypoint detectors have been developed but also binary descriptors such as the BRIEF (Binary Robust Independent Elementary Features, (Calonder et al. 2010)), the BRISK (Binary Robust Invariant Scalable Keypoints, (Leutenegger et al. 2011)) or the FREAK (Fast REtinA Keypoint, (Alahi et al. 2012)) are becoming more and more popular in machine vision. Instead of storing a vector of orientation histograms, they compare a well defined set of pixel intensities in the neighborhood of the according keypoint which results in a Boolean vector. These descriptors are not only much faster to compute but also faster to match because their similarity can be measured using the Hamming distance i.e. the number of entries in the Boolean vector that don't match. These approaches can also be made rotation invariant e.g. in ORB (Oriented FAST and Rotated BRIEF, (Rublee et al. 2011)) and have proven to be powerful in robotics as well as augmented reality applications.

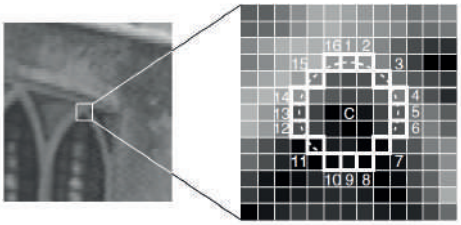


Figure 4.3: FAST corner detection: the intensity of pixel 1-16 are compared with the one in the center C. Reprinted from (Rosten et al. 2005).

4.1.4 Detection of Geometrical Primitives

Many machine vision applications operate in geometrically constrained environments and simple objects such as lines, circles or rectangles have to be detected. For such applications a set of powerful algorithms have been developed that do not rely on keypoints.

4.1.4.1 Hough Transform

The Hough transform has been proven to be one of the most powerful tools (Illingworth et al. 1988) to detect geometrical primitives. Pixel coordinates get mapped into a parameter space that represents the probability of geometric shapes with the according parameters. For line detection, a point in the pixel space gets projected onto the parameters of all possible lines (with angle ρ and distance r from the origin) that go through it. By searching for the maximally scoring parameters, the most probable (set of) shape(s) is determined.

4.1.4.2 LSD: Line Segment Detector

Another approach to perform line segment detection is the Line Segment Detector LSD (Gioi et al. 2010; Gioi et al. 2012) which works more on local features than on global parameters.

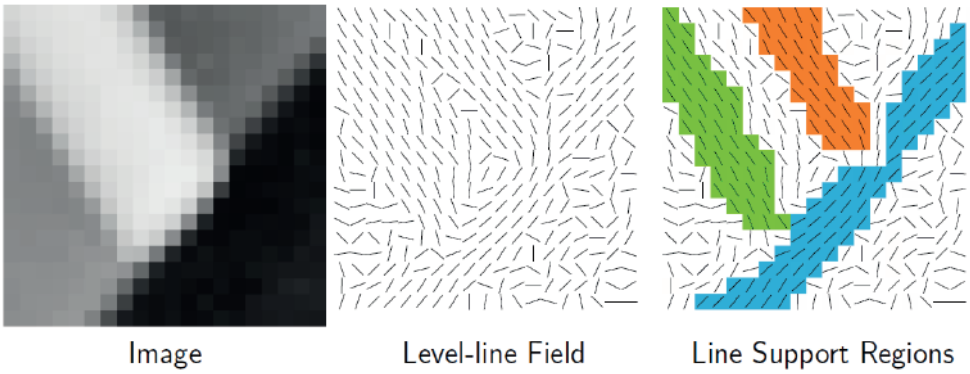


Figure 4.4: The creation of the support regions in the LSD algortihm. Reprinted from (Gioi et al. 2012).

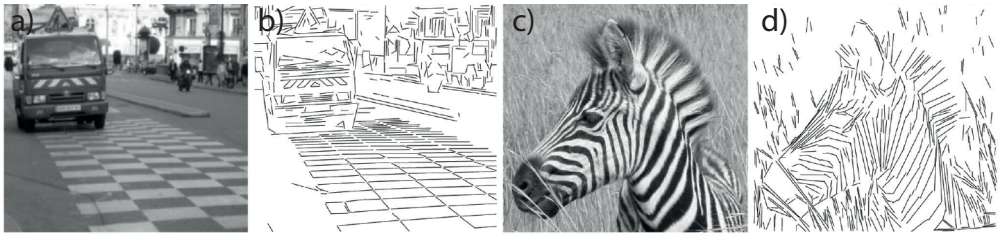


Figure 4.5: The output of the LSD algorithm a),c) original grayscale images b),d) line segments detected with LSD. Reprinted from (Gioi et al. 2012).

Line-Support Regions
The first step in the LSD is to compute the level-line field of the image: for each pixel the intensity gradient is computed and the level line corresponds to a line which runs perpendicular to the gradient. In a next step the line-support regions are determined: connected sets of pixels that share the same level line orientation i.e. spatial gradient orientation (as shown in Fig.4.4). For each line-support region a bounding box is placed so that it fits the support region. Too small line segments or ones with badly aligned points are sorted out. To sort out line segments that are noise-generated, a statistical test is performed. The resulting line segments describe arbitrary images as a set of line segments and thereby parametrize them (Fig.4.5)

4.2 Temporal Contrast

Many event-based machine vision algorithms have been developed for the output of the dynamic vision sensor which encodes temporal contrast. In the following the mathematical properties of temporal contrast are outlined to explain and justify some of the algorithms described in the following.

4.2.1 The DVS Events

The DVS computes temporal contrast i.e. discrete changes in the log light intensity at a pixel. The original paper(Lichtsteiner et al. 2006b) expresses temporal contrast C^t in the following way:

$$C^t = TCON = \frac{1}{\hat{I}(t)} \frac{\partial \hat{I}(t)}{\partial t} = \frac{\partial (ln(\hat{I}(t)))}{\partial t} \tag{4.2}$$

where $\hat{I}(t)$ represents the photocurrent. Under the assumption of a fixed quantum efficiency, the photocurrent is proportional to the illuminance at the pixel $E_v = c \cdot \hat{I}$. Using a log compression allows simplifying the temporal contrast expression and working in log illumination space

$$I = ln(E_v) = ln(c \cdot \hat{I}) \tag{4.3}$$

$$C^t = \frac{\partial I}{\partial t} \tag{4.4}$$

where I is an expression for log light intensity i.e. log-illuminance. The DVS events encode discrete steps in the log illumination

$$M(Ev_i) = \begin{cases} \theta^k_{OFF} & , Pol(Ev_i) = 0 \\ \theta^k_{ON} & , Pol(Ev_i) = 1 \end{cases} \tag{4.5}$$

which allows to infer temporal contrast from the timing (equivalently for OFF events)

$$C^t = \frac{\delta I}{\delta t} = \frac{\theta^k_{ON}}{T(EV^k_j) - T(EV^k_{j-1})} \tag{4.6}$$

4.2.1.1 The Optical Flow Assumption

The temporal contrast encoded by the DVS events i.e. the changes in illuminance of the pixel can have different sources:

- *Changes in Reflectance:* The ratio of reflected light over absorbed light changes in some parts of the observed scene. This can be caused by changes in the surface but in most cases it is caused by motion of surfaces with spatial contrast over the image plane.
- *Changes in Luminous Intensity:* The light falling onto the observed scene changes. This is the case with flickering light sources or changing weather. The according events are generated all over the scene.
- *Noise:* The reset transistor source/drain diffusion in the DVS leaks current to the positive supply so that the pixel is constantly producing fake ON events at a low frequency of about 0.1Hz. Thermal noise can also trigger events which can be seen especially under dark conditions.

The events don't carry any dedicated information on what caused them so assumptions have to be made. In active light scenarios such as active marker tracking or structured light, the timing information can be used to filter out events that are not caused by the active light source. In passive light scenarios, the *optical flow assumption*, as it is called in the following, allows treating the events without additional timing information. The assumption includes the following:

- The lighting of the scene is constant or changing at a negligible rate.
- The reflectance pattern of the objects in the scene is constant or changing at a negligible rate.
- The noise events occur at a negligible rate or can be filtered out.
- The majority of events are caused by structures with spatial contrast moving relative to the sensor.

This is called the *optical flow assumption* because under given constraints, the source of most events can be described by the optical flow equation (Horn et al. 1981; Horn et al. 1993).

4.2.2 Optical Flow Equation

The optical flow equation which was also investigated in relation to the DVS in (Benosman et al. 2012; Benosman et al. 2014; Tschechne et al. 2014) and can be formulated as follows:

$$\frac{\partial \hat{I}}{\partial x}\delta x + \frac{\partial \hat{I}}{\partial y}\delta y + \frac{\partial \hat{I}}{\partial t}\delta t = 0 \tag{4.7}$$

or reformulated to:

$$\frac{\partial \hat{I}}{\partial t} = -\frac{\partial \hat{I}}{\partial x}v_x - \frac{\partial \hat{I}}{\partial y}v_y \tag{4.8}$$

where $\hat{I}$ is the pixel intensity value (photocurrent), $\frac{\partial \hat{I}}{\partial x}$, $\frac{\partial \hat{I}}{\partial y}$, $\frac{\partial \hat{I}}{\partial t}$ the according spatial and temporal derivatives and v_x/v_y are the velocity components by which a structure with spatial contrast is moving relative to the sensor image. If the logarithm is taken, the temporal contrast becomes a function of the spatial contrast and the velocity:

$$\frac{\partial ln(\hat{I})}{\partial t} = -\frac{\partial ln(\hat{I})}{\partial x}v_x - \frac{\partial ln(\hat{I})}{\partial y}v_y \tag{4.9a}$$

$$C^t = -\frac{1}{\hat{I}}\frac{\partial \hat{I}}{\partial x}v_x - \frac{1}{\hat{I}}\frac{\partial \hat{I}}{\partial y}v_y \tag{4.9b}$$

$$C^t = -\frac{1}{\hat{I}}(V^T \cdot C^s) \tag{4.9c}$$

where C^t is the temporal contrast, C^s is the spatial contrast in the form of derivatives and V is the velocity vector:

$$C^s = \begin{pmatrix} \partial \hat{I}/\partial x \\ \partial \hat{I}/\partial y \end{pmatrix} \tag{4.10a}$$

$$V = \begin{pmatrix} v_x \\ v_y \end{pmatrix} \tag{4.10b}$$

So under the optic flow assumption, the event rate $\delta ln(\hat{I})/\delta t$ is proportional to the spatial contrast in a scene and velocity by which it is moving relative to the camera. This relation can not only be used to infer the optic flow but it also allows tracking spatial contrast structures as shown in the *ELiSeD* study.

4.3 Event-Based Algorithms

When designing and implementing an algorithm for event-based sensors, there are multiple decisions to be made. These decisions are very critical for the performance of the algorithm and should be made before their implementation. In the following the most important aspects are discussed and trade-offs are shown. The basis for most of these considerations was already set with the first jAER filters(Delbruck 2008). It is assumed that the algorithms are designed for conventional serial processing platforms (CPU's).

4.3.1 Processing Routine

When working with asynchronously acquired events in a clocked processing system, these events can be processed at different points in time. Depending on the input data, the computed output and the computational complexity, different processing routines should be used.

4.3.1.1 Regular

The incoming events can principally be accumulated in a memory representing the input space and the data in this memory is then processed in a regular frequency. For the DVS output this corresponds to accumulating all the events during a fixed time period to a temporal contrast frame which is then processed. This approach makes sense if the output of the algorithm is polled with a regular frequency and if the computation per event is computationally expensive. The computational L load can be computed by:

$$L = f_p \cdot S_M \cdot i_M \tag{4.11}$$

where f_p is the processing/polling rate at which the buffered events are processed, S_M is the size of the memory required to represent the event address space and i_M are the number of instructions to be performed for each entry in the memory.

4.3.1.2 Event-Based

The approach that creates the lowest latency is the event-based one: the output of the algorithm is updated upon the arrival of each event. The drawback of this approach is that the computational load grows with the number of events while the load in the case of regular updates is constant:

$$L = \bar{f}_E \cdot i_E \tag{4.12}$$

Where f_E corresponds to the average event rate and i_E to the number of instructions.

4.3.1.3 Packet-Based

For some computations that have to be updated from time to time, it makes sense to perform these computations at the beginning or end of an event packet. In jAER updates which do not directly depend on the event input can be performed before or after a programmable set of events (packet) gets processed. This allows to run costly processing steps at a regular interval without the requirement to update them for every event. Especially for processes that have to happen independent of the event input e.g. removing objects that did not get an event input, this is a valuable option.

4.3.1.4 Comparison

Interestingly it is not necessarily smarter to update the output in an event-based manner. In the case of an operation that is the same for an event and for a memory entry, the regular update scheme leads to less computational load than the event-based if $f_E > f_p \cdot S_M$. This can be illustrated in following example: performing an event-based convolution requires the same amount of instructions as a pixel-based convolution. Given that the sensor has a resolution of 240x180 (43200) and the output of the convolution is required to be updated at a frequency of 30Hz, a frame-based convolution has less computational load if the event rate exceeds 1.3Meps.

4.3.2 Temporal Relations

In conventional computer and machine vision, the frame is the basis of all computation and most operations rely on neighboring pixels within that frame and/or a number of frames before or after that frame. While spatial relations for events can be handled similar as in frames, there are several options how time should be treated. The fundamental question is which past events should affect the output of the computation and how much. The following describes a few methods how temporal relations among events can be handled but there might be more as well as combinations of these methods.

4.3.2.1 Regular Time Window

The simplest way to treat time is to slice the event stream it into regular time windows with a given frequency f_R. By accumulating the event values, pseudo-frames can be created which can then be processed by a regular frame processing approach.
For event-based processing routines this approach is less suited because it can arbitrarily separate two events into different windows. In addition this scheme adds an average latency of $1/2f_R$ and the output is dependent on the velocity of a given stimulus because faster objects produce more events per time and therefore more events per temporal window. This approach is rarely beneficial except for statistical measurements and event rendering.

4.3.2.2 Sliding Time Window

One way to avoid the random separation of events across time windows, is to let the window slide with the incoming events so that the time of the latest event marks one end of the window and this time minus a fixed amount ϵ marks the other. Upon the arrival of an event, all events older than ϵ are removed from the output. This approach defines clearly which events should affect the output but it has the same drawback as the regular time window: it is velocity dependent. In addition it requires dynamic memory allocation: an unknown amount of events have to be

stored somehow so that they can be removed as soon as they drop out of the window.

4.3.2.3 Decaying Time Window

One of the drawbacks of the sliding and regular time window is that they treat recent events with the same weight as old ones. This leads to lag in the output because the result is averaged over all events in the window. Another problem is that an unknown number of events has to be stored which requires an unknown amount of memory. This implies that expensive dynamic memory structures are required. One way to circumvent this problem is to weigh the event's effect with a temporally decaying value. In most cases this decay is either linear which is easy to implement or exponential which is more bio-inspired (synaptic dynamics) and has less lag. In some algorithms the exponential decay is approximated with a bit shift operation which is much cheaper to compute. The decaying time window can computationally be very costly depending on when the decay is computed:

- If the output of the algorithm can change even when no events are arriving, than the internal states and the output of the system have to updated on a regular basis. This requires costly iterations and often the computation of the expensive exponential function.

- If the output of the system can only change upon the arrival of an event (such as in a so-called integrate-and-fire neurons), the decay can be performed just before the event is added.

One advantage compared to the windowing approaches is that the events do not have to be stored but only the most recent result of the according value. A drawback is that each event has only a temporally limited effect on the output of the algorithm.

DVS Data

In the case of change encoding temporal contrast events such as in the DVS, this approach leads to algorithms with a velocity dependent output. This can be illustrated with the optic flow equation: if the optic flow assumption holds true and it is further assumed that the spatial contrast in a scene is constant $C^s = c$, then the event rate $R_E = n/\Delta t$ (n being the number of events arriving in a time window Δt) becomes a function of the velocity:

$$C^t = -\frac{1}{\hat{I}}(V^T \cdot C^s) \tag{4.13a}$$

$$\frac{\Delta I}{\Delta t} \cong \frac{n \cdot \theta}{\hat{I}\Delta t} = -\frac{1}{\hat{I}}(V^T \cdot c) \tag{4.13b}$$

$$R_E = \frac{n}{\Delta t} \cong -c \cdot V^T \tag{4.13c}$$

This implies that the number of events which are summed and decayed is proportional to the velocity. Given the delay function $D(t)$ is the same for all events, its support is also constant for all events $supp(D(t)) = q$. If q is smaller than $1/R_E$ the decays of the events do not overlap each other and can thereby not affect each other i.e. no information can be integrated over events. If the supports are overlapping, their overlap and therefor influence on each other and the output of an arbitrary function is dependent on the event rate i.e. velocity. This holds also true for the fixed and sliding time windows where $D(t) = 1$ and $supp(D(t)) = 1/f_R$
This can be illustrated with following example: a slow moving object generates about the same number of events as a fast one by the time it completely passed the field of view of a DVS so the sum of events is constant. But the sum of events falling a specific time window depends on the speed of the object. But even though this is an disadvantage in many algorithms, it is used in multiple algorithms that are not intended to be velocity dependent (such as the rectangular cluster tracker in jAER). These applications might at some point be translated into a ring buffer approach for better stability.

4.3.2.4 Ring Buffer

If the decision on which events should be relevant for the computation is based on a temporal criterion, this necessarily leads to velocity dependence. But velocity dependence is unwanted in

most applications such as tracking. Another way to chose the relevant events is to neglect the absolute time and rely on their temporal order. Instead of computing with all events from the last X ms, only the latest X thousand events are relevant for the output. Implementing such a scheme based on the ordering of the events can be done with a ring buffer where each new event replaces the latest event in the buffer.
While computation based on temporal criteria leads to velocity dependence, the ring buffer approach depends on the spatial contrast in a scene. When a scene contains a lot of spatial contrast, the ring buffer might not be capable of buffering an event from each edge. But when the buffer is too big, the computation relies to much on old events which leads to lag.
There are multiple strategies to tackle the buffer size problem:

- The spatial event distribution allows to infer the amount of spatial contrast in a scene and the buffer size can be adapted accordingly.
- If the spatial structure causing the events is tracked and the event correspondence problem solved, the number of events in the buffer which are caused by the same spatial structure can be assessed. By setting a desired value for how many events per structure should be in the ring buffer, a controller (e.g. PID controller) can be implemented.

4.3.3 Memory Management

A critical aspect when implementing event-based algorithms is the memory management because it affects the memory usage of the software and its performance. For efficient memory management it is important that information can be accessed with as few searches as possible. This can be achieved if the information is stored under a meaningful index: spatial information should be stored in a spatial array where indexes correspond to spatial locations and temporal information in a temporally ordered array. These two storage modalities can be interlinked using pointers to the other storage units. To get the information about pixel x, y it should not be necessary to iterate over the event buffer but the information should be found in an array accessible with indexes x, y [1]. In the same way it should be easy to find the latest events in a temporally ordered buffer instead of iterating over a full picture.

4.3.4 Delta Ring Buffer

Inspired by the integral images in conventional machine vision (Viola et al. 2001), the delta ring buffer was developed as a powerful tool to perform efficient event-based computation. The basis of this concept is to compute only the delta (difference) by which an event affects an internal state or output and store these deltas in a ring buffer. Alg.1 shows how such a delta ring buffer method is structured: For each a new event ev of the latest packet pkt, the oldest delta $oldD$ is subtracted from the result ("updateOutput" method), the new delta computed ("computeDelta" method), added to the result, stored in the ring buffer buf and the pt is increased to point to the new oldest event. After any step the output out can be read out or used for further processing. For a better understanding of what

Algorithm 1 Delta Ring Buffer

```
1:  procedure FilterPacket(pkt)
2:      for ev:pkt do
3:          oldD ←buf[pt]
4:          out ←updateOutput(-oldD,out)
5:          newD ←computeDelta(ev,buf)
6:          buf[pt] ←newD
7:          out ←updateOutput(newD,out)
8:          pt ←pt+1
9:          if size(buf) =< pt then
10:             pt ←0
11:         end if
12:     end for
13: end procedure
```

the "updateOutput" and the "computeDelta" methods do, the delta ring buffer is illustrated

[1] 1D arrays are significantly faster than 2D array so a method to convert 2D coordinates to 1D indexes should be used

in a simple example. Given the average timestamp over the latest 1000 events should be determined (and "TS" is returning the timestamp of an event), the delta ring buffer allows to do this with an addition, a subtraction (addition of a negative number) and a divison per event as shown in Alg.2 The biggest advantage of the

Algorithm 2 Example Methods

```
1: procedure size(buf)
2:     return 1000
3: end procedure

4: procedure updateOutput(dta,out)
5:     return out+dta
6: end procedure

7: procedure computeDelta(ev,buf)
8:     return TS(ev)/size(buf)
9: end procedure
```

delta ring buffer approach is that the output is updated on each event which preserved the low latency in the computation without the requirement of iterating over multiple events. The delta ring buffer is used in the *Event-Based Keypoints*, the *Laser Line Extractor* and the *ELiSeD* described in the studies below.

4.4 Event-Based Software

Handling event-based data requires dedicated software. The processing architecture of this software depends on how the events are communicated. In most cases such as DVS and DAVIS, this communication is performed through USB which requires that the events are sent as packets. The processing software discussed in the following is built around processing these packets.

4.4.1 jAER

The Java-based framework for address event representation (jAER) is a user-friendly (Garcia Franco et al. 2013) open-source software package developed by T. Delbruck, supported by multiple authors and maintained by iniLabs. It allows to display and log event-based data as well as a fast implementation of simple event-based algorithms. It hosts multiple already existing algorithms and processing routines and allows to configure several event-based chips.
The processing routine is initiated by the so-called hardware interface which performs the timestamp expansion and translates incoming USB packets into raw event packets. These raw events then get translated into event objects by the event extraction method of the according chip class. The events are then processed in so called event filters which get a packet of events and then iterate over them to perform some arbitrary computations.

4.4.2 cAER

To allow for embedded event-based processing without the overhead of a full Java project, the C-based framework for address event representation (cAER) was developed by L. Longinotti (Longinotti 2014). In contrast to jAER, cAER is intended to be as "slim" as possible and optimized for performance. For this reason the program is structured very modular and instead of allowing to configure the processing routine and interface structure in runtime, the processing architecture is configured before the compilation and only certain parameters can be adjusted during runtime through a configuration server.

In contrast to jAER, cAER is not built around a graphical user interface (GUI); it can even be detached from the terminal it was started from and run in the background as a demon process. Any cAER implementation is built around modules: *inputmodules* acquire events from a sensor, *processingmodules* process the information in the desired way and *outputmodules* send out the required information into a file or through a TCP or UDP port to other programs/processes. One or multiple main loops define the order in which these modules are called and what type of information they process.

4.5 Event-Based Localization

Since the key advantages of DVS are it's low latency and high temporal resolution, it is suited for high-speed localization tasks. For this reason, the first algorithms developed were used for localization.

4.5.1 Tracking

4.5.1.1 Convolution-Based Tracking

The first algorithm applied to the dynamic vision sensor data was a hardware implementation of a convolution-based tracker. The CAVIAR project in which the DVS was developed, was built around a asynchronous, event-based, full sensory-motor demonstrator that implemented a tracker (Serrano-Gotarredona et al. 2005; Serrano-Gotarredona et al. 2009). The events of the DVS are sent to a convolution chip that detects shapes, a winner-take-all chip to find the best matches and a tracking circuitry as well as learning classifier chip that can learn spatio-temporal patterns. Even though this was run on dedicated hardware it could also have been implemented in software.

4.5.1.2 Geometrically Constrained Tracking

The first commercial application of the dynamic vision sensor was detecting cars on a lane, counting them and estimating their velocity. The spatial constraints in this setup allow simplifying the problem: Under the assumption that the scene is uniformly illuminated and the sensor mounted in a fixed position, the events must be generated by cars moving along their lane. This allows to detect and track cars depending on the temporal activity and determine their velocity (Litzenberger et al. 2006a; Litzenberger et al. 2006b; Bauer et al. 2007; Litzenberger et al. 2007b; Litzenberger et al. 2007a).
Another tracking algorithm that uses geometrical constraints was used to extract lines and lanes on a road to control an RC (Brandli 2008). The underlying algorithm detects peaks in the horizontal event activity to detect and model the lines along which the car is driving.

4.5.1.3 Event Cluster Tracking

On the same platform that was used for traffic surveillance also another tracking algorithm was implemented: a general purpose tracker for objects and people (Litzenberger et al. 2007a). In contrast to the vehicle tracking, this algorithm clusters events without using geometrical constraints of the setup by using a threshold-ed nearest neighbor criterion: Each incoming event is assigned to the closest cluster tracker and the tracker position is updated using a weighted average. If the closest cluster is too far away, a new cluster tracker is created.
The same principle combined with a Kalman filter was used to track the wings of fruit flies (Cardinale 2006).
A similar way of assigning events to a ball or an arm tracker was used in the so-called robo-goalie that estimates the motion of a ball shot at a goal to then defend it using a 1 DoF arm (Lang 2007; Delbruck et al. 2013).
The event cluster approach was later on improved by using stereo vision to assign depth to the events and focus the clustering on a certain distance of interest.(Schraml et al. 2010c; Schraml et al. 2010d)
This type of event cluster tracker has also been implemented in a hardware description language (Gomez-Rodriguez et al. 2010; Gomez-Rodriguez et al. 2011).

In another application such a cluster based tracker was used to track particles and estimate their motion (Drazen et al. 2011) and it has the potential to characterize cell phenotypes in impedance spectroscopes (Haandbaek et al. 2011).

4.5.1.4 Hough Tracker

By applying the Hough transform to the DVS events, simple shapes such as the circular iris can easily be located and tracked (Gisler 2007).
The Hough tracker can be simplified if geometrical constraints are taken into consideration. The algorithm that controls the DVS-based pencil balancing robot only evaluates the data of two distinct horizontal regions to find the parameters that describe the position and orientation of the pencil (Conradt et al. 2009a; Conradt et al. 2009b; Conradt et al. 2009c).
Another application of the Hough circle tracker is used track circular structure underneath a microscope (Ni et al. 2013; Ni 2013).
The Hough tracker can also be used for a fast and computationally cheap camera pose estimation for micro-aerial vehicles (Mueggler et al. 2014).

4.5.1.5 Shape-Based Tracker

If the shape of an object is known, the event stream of the DVS allows to infer the most probable translation and rotation of given object.
One approach to perform such shape based tracking is using the active appearance model algorithm (Tureczek 2008).
Another powerful method to do so is the iterative closest point algortihm (ICP). This algorithm allows for instance to track the position of a gripper underneath a microscope to generate haptic feedback to the operator (Ni et al. 2012; Bolopion et al. 2012; Ni 2012).
The undocumented "Pig Tracker" developed by M. Cook et al. which can be found in jAER is another example of a shape-based tracker.

4.5.1.6 Active Marker Tracking

One advantage of the DVS is its high temporal resolution which allows to detect temporal patterns such as the blinking of a light emitting diode (LED). A highly useful cue to estimate the blinking frequency of a pixel is to measure the time between the ON-OFF and OFF-ON transitions i.e. the time the first event of the according polarity arrives. Together with spatial motion assumptions this leads to a stable tracker(Muller et al. 2011).
But the temporal ON-OFF transitions also allow identifying it by its frequency or its blinking pattern(Hofstetter 2012).
A stereo setup combined with a neural network also allows tracking the position of a LED in three dimensions(Müller et al. 2012).
Multiple LEDs allow inferring all six degrees of freedom of a tracked system such as a microaerial vehicle(Censi et al. 2013).

4.5.2 Depth Estimation

4.5.2.1 Stereovision

Already the pioneering work of M. Mahowald included winner-take-all analog circuits to compute stereo-correspondences (Mahowald 1994a) but the first event-based stereovision algorithms were only developed with the dawn of the DVS. The first but unfortunately rarely credited approach to match event streams from two event-based sensors was based on the Marr-Poggio stereo matching algorithm(Marr et al. 1979) enforcing matches for a single frontal object (Hess 2006). The same work then generalized this approach to allow for arbitrary shapes. Interestingly this work already included epipolar geometrical constraints, event-timing, polarity and event "orientation". A similar approach to this was implemented with a more sensitive DVS and dedicated hardware to compute the orientation using Gabor filters (Serrano-Gotarredona et al. 2013a; Camunas-Mesa et al. 2014).
The team from the Austrian Institute of Technology that also developed the traffic and people tracker followed a different approach to solve the stereo-correspondence problem: accumulating the activity and performing a convolution along the epipolar geometry (Schraml et al. 2010c; Schraml et al. 2010b; Schraml et al. 2010d; Belbachir et al. 2011b; Belbachir et al.

2011a; Kohn et al. 2012b). They have also developed a system for the ground truth evaluation of their systems (Kogler et al. 2013).
R. Benosman further invested the epipolar constraints for event-based sensors (Benosman et al. 2011) and used them for stereo event matching(Rogister et al. 2012). This approach was then extended to N-ocular stereovision setups (Carneiro et al. 2013).
Another approach to estimate depth is based on the relation between the focus of the optics and the amount of events (Domi?nguez-Morales et al. 2012) as well as epipolar geometry (Dominguez-Morales et al. 2013).
Another way of approaching the stereo-matching problem is to store the most recent events as activity in a neural network (Piatkowska et al. 2014).

4.5.2.2 Structured Light

The highly resolved temporal resolution of the DVS is not only useful in active marker tracking and identification but also for active sensing. Pulsing a light pattern onto a scene with a known timing allows separating events caused by the pattern from motion induced events. By applying a temporal filter, the DVS can be used for tasks such as profile extraction(Brandli et al. 2014b) or potentially also on more complicated patterns for full depth maps such as the ones used in Microsoft Kinect.
Surface reconstruction with the help of a DVS and a pulsed laser line is discussed in more detail in the according study section on page 79.

4.5.3 Gesture Detection and Recognition

If the DVS is fixed, it captures the motions around it and represents it sparsely which can be used for gesture detection and recognition because the background information is already subtracted on the sensor.
Kohn et al. directly used a Hidden Markov Model (HMM) on 8x8 down-sampled, 40ms frames of an overlay of two spatially separated sensors to classify gestures (Kohn et al. 2012b; Kohn et al. 2012a). Even though this approach uses a stereo setup, it does not use the depth information apart from the disparity in the overlay images.
Lee et al. on the other hand use leaky integrate-and-fire neurons that allow responding only to events with a certain depth i.e. disparity. The activity of these neurons with a specific depth-depending receptive fields is then clustered, tracked and motion features are extracted. The HMM is only used to determine the sequence of the motion features (Lee et al. 2012a; Lee et al. 2012b; Lee et al. 2014).
Another approach of gesture detection is based on spatio-temporally tuned filters (Tschechne et al. 2014).

4.5.4 Optic Flow

Due to its simple representation of temporal contrast, the DVS allows to infer the optic flow at a pixel if the spatial derivative is known.
The optic flow can either be computed by solving the optic flow equation for each event (Benosman et al. 2012), using a neural network (Orchard et al. 2013), a derivative on the time surface(Benosman et al. 2014) or filters with spatio-temporal kernels(Tschechne et al. 2014).
The optical flow is a useful feature to determine for instance the time to contact in mobile robots (Clady et al. 2014).

4.5.5 Region of Interest

By only signaling information on the parts of a visual scene that are moving, the events are useful cues to direct the attention and focus of processing towards this region (Rea et al. 2013). In combination with a conventional camera or in a DAVIS, this field still offers several interesting applications in which the events can serve as guides for the frame readout to suppress redundancy. The events can be used to control the time and region of the intensity readouts to focus on the interesting aspects of a visual scene. Unfortunately there is no published work in this field.

4.5.6 Simultaneous Localization and Mapping (SLAM)

A key problem in mobile robotics is to map the unknown environment in which a robot is and localize it on this map. The sparse data representation of the DVS allows to simplify this simultaneous localization and mapping (SLAM). The Conradt group has developed an algorithm that allows to solve the SLAM problem based on the DVS output (Weikersdorfer et al. 2013; Hoffmann et al. 2013). Due to the complexity of this problem, their particle filter-based approach is restricted to applications in 2D navigation.

4.5.7 Visual Odometry

The temporal contrast information allows inferring camera motion that caused it. Measuring the motion of the camera or the camera platform based on visual information is called odometry. Censi and Scaramuzza (Censi et al. 2014) infer the camera motion by aligning the output of a DVS with the images of a conventional frame-based camera. The frames allow them to compute the spatial derivatives and the temporal contrast events allow to infer how the system must have moved to cause these events.

4.5.8 Topology

The temporal information in the event-stream of a pixel allows comparing the similarity among pixels and thereby infer the pixels neighborhood relations just from the event stream. This allows to learn the sensors topology which has been demonstrated in (Schrag 2008) and (Boerlin et al. 2009). These neighborhood relations also allow to learn motion features in a self-organized way (Koeth et al. 2013).

With the help of an inertial measurement unit (IMU), it is also possible to remap the event coordinates and thereby stabilize the visual input against shaking (Delbruck et al. 2014).

4.6 Event-Based Identification

Even though the DVS has a low spatial resolution and does not report absolute intensities, it has been employed in various identification and classification tasks.

4.6.1 Event Detection

The sparse nature of the DVS output allows to simplify the detection of events or objects.
The DVS is used for fall detection in the context of elderly care where statistical measures are used to classify a fall (Fu et al. 2008). This approach has been improved using a neural network (Sulzbachner et al. 2012) and depth information (Belbachir et al. 2011b; Belbachir et al. 2012a)
The use of near infra-red (NIR) sensitive event-based sensors (all standard photodiodes are sensitive to NIR) allows to detect the presence of flames using a simple threshold approach (Lenero-Bardallo et al. 2013).

4.6.2 Classification and Recognition

The spatio-temporal content of the DVS event-stream allows to classify and recognize the sources or objects they originate from.

4.6.2.1 Primitive-Feature-Based Classification

Due to the constraints in a traffic surveillance setup where the sensor is looking at cars driving along fixed lanes, it is simple to count them by thresholding the event activity in a certain region of interest (Litzenberger et al. 2007b).
An early heuristic approach to event-based classification uses manually tuned parameters to classify traffic surveillance data into the classes car or truck at day and night (Gritsch et al. 2008; Gritsch et al. 2009). This approach has been further developed to also classify and count freely moving pedestrians and cyclists (Belbachir et al. 2010b; Schraml et al. 2010a).

In another setup using an event-based dual line sensor, cars are recognized using basic features (Belbachir et al. 2007). This approach was further developed to use features derived from convolutions with geometrical primitives such as circles (Belbachir et al. 2011c).

4.6.2.2 Orientation-Based Classification

A bio-inspired approach on object classification follows the HMAX model of the visual cortex. Using Gabor filters, this approach extracts orientation features (line segments) and uses them to classify postures (Chen et al. 2009; Chen et al. 2012). Unfortunately this approach bins the event-based output into frames and thereby loses some of the advantages of event-based data. This shortcoming has been overcome by a pure event-based implementation (Zhao et al. 2014).

4.6.2.3 Texture Recognition

With the help of Gabor convolutions on the event stream it is possible to recognize and classify textures (Perez-Carrasco et al. 2010).
By using Fourier transform and polarization filters, the DVS can be used to infer properties of the street on which a motorbike is driving (Dankers 2014).

4.6.2.4 Convolutional Networks

By using convolutional networks, event-based sensors can be used to recognize poker card symbols (Perez-Carrasco et al. 2013).

4.6.2.5 Deep-Belief Networks

A very promising approach of using deep belief network structures has been used to classify digits in real-time using auditory and visual event-based data (O'Connor et al. 2013).

4.7 Event-Based Reconstruction

Since the DVS events carry information on the temporal contrast, they can be used to recon-

struct the spatial contrast and the intensity in a scene.

4.7.1 DVS-based Reconstruction

The first event-based reconstruction was using a mechanical shutter and the events were just integrated (mentioned in (Lichtsteiner 2006; Brandli et al. 2014c)). A more elaborate approach is using interacting maps that are connected through factor graphs which infer the rotational motions of the camera, the spatial contrast and the light intensities in the scene from just the DVS events (Cook et al. 2011).
A similar approach that does not use interacting maps but solves a similar optimization problem (Kim et al. 2014) unfortunately does not refer to the pioneering work of M. Cook et al.

4.7.2 360° Panoramic View Reconstruction

When an event-based line sensor in mounted on a rotating platform it allows to capture a 360° panorama of events (Belbachir et al. 2010c). The quality of the panorama depends strongly on the rotation speed (Belbachir et al. 2012b). These spatial contrast images can be converted into grayscale images by spatial high-pass filtering, calibrating the ON/OFF ratio and integrating the events (Belbachir et al. 2014). The quality can be assessed using a dedicated tool (Graf et al. 2013).

4.7.3 ATIS-based Reconstruction

In the ATIS, the background activity noise triggers random intensity readouts which can be used to gradually decompress a scene using a matching pursuit algorithm (Orchard et al. 2012). In the ATIS not only the time between the first and second intensity readout carries information about the light intensity but the timing between all events can be exploited to reconstruct the scene (Orchard et al. 2014a).

4.7.4 DAVIS-based Reconstruction

The DAVIS frames can be used as a ground truth to reconstruct the observed scene based on the event-based temporal contrast (Brandli et al. 2014c). The algorithm is described in the section "DAVIS Decompression Study" on page 83.

4.8 Event-Based Keypoint Features

To infer absolute light intensities from the DVS events without an estimation of the camera motion and depth estimates for the pixels is difficult, but events carry information on the relative light intensities. Most keypoint detectors and descriptors operate on spatial contrast i.e. relative light intensities and so it should be possible to use the events to detect and describe conventional keypoints. This approach was investigated in a Bachelor thesis reported in the section "Event-Based Keypoints Study" on page 91.

4.8.1 Unsupervised Features

By presenting event-based information to networks with particular learning rules, they organize themselves so that the neurons respond to very distinct spatiotemporal features (Bichler et al. 2011; Bichler et al. 2012a; Bichler et al. 2012b; Roclin et al. 2013).

4.8.2 Features for Robotic Vision

Visual features are of high importance in the field of robotics. They can either be based on event clusters (Wiesmann et al. 2012) or generated from neurons (Lagorce et al. 2013).

4.8.3 Image Filtering

Image filtering plays a crucial role in frame-based vision algorithms for instance to detect features. Event-based sensors can improve the performance of such approaches by guiding the image sampling and updating (Ieng et al. 2014).

4.9 Event-Based Vision Hardware

Apart from software implementations, there have been several hardware implementations of event-based vision processing.

4.9.1 Event-Based Convolution Hardware

Convolutions are powerful tools to detect features in an input space and this can also be exploited for event-based data. Already in the CAVIAR project from which the DVS originated, dedicated aVLSI convolution hardware played a crucial role (Serrano-Gotarredona et al. 2005; Serrano-Gotarredona et al. 2009) and it was continuously developed further (Serrano-Gotarredona et al. 2006; Camunas-Mesa et al. 2008; Serrano-Gotarredona et al. 2008; Camuñas-Mesa et al. 2009; Camunas-Mesa et al. 2011; Camunas-Mesa et al. 2012).
Other approaches to implement event-based convolutions are realized in graphics processor units (GPUs) (Nageswaran et al. 2009) or in field programmable gate arrays (FPGAs) (Linares-Barranco et al. 2010; Rivas-Perez et al. 2010).
Convolution chips have been shown to perform event-based character recognition (Perez-Carrasco et al. 2008). Character recognition was also shown on dedicated FPGA processing architectures (Neil et al. 2014).

4.9.2 Bio-Inspired Processing Hardware

Apart from convolution based processing, event-based visual information can also be processed on multi-neuron platforms (Vogelstein et al. 2007). Orientation cues of the input data can be extracted using dedicated hardware such as a PCI-AER interface board (Chicca et al. 2006; Chicca et al. 2007). A bio-inspired model of selective attention on a VLSI chip has been implemented by Bartolozzi and Indiveri (Bartolozzi et al. 2009) and further characterized by Sonnleithner (Sonnleithner et al. 2011; Sonnleithner et al. 2012). In this multi-chip system the event-based output of the DVS is performing a winner-take-all operation to focus on the most salient part (region of interest) of a scene.
Other event-based attention models were implemented on the SpiNNaker platform (Galluppi et al. 2012) or the iCub platform (Rea et al. 2013).

4.10 Study Event-Based Structured Lighting

A key advantage of event-based vision is its access to the time dimension that allows detecting temporal patterns in the visual input. In active marker scenarios it is the knowledge about the temporal blinking pattern that allows increasing the signal-to-noise ratio and thereby reliably detecting and tracking the marker. But since the remote blinking pattern phase is not known, the phase information of the signal has to be recovered (Hofstetter 2012). If the light pattern originates from the same location as it is detected, the phase information can be transmitted through a direct synchronization connection. Together with geometrical constraints, this allows to recover depth information as described in the following.
This work was implemented by T. Mantel in a semester project co-supervised and based on ideas by C. Brändli and M. Hutter / M. Höpflinger from the Autonomous Systems Lab, ETH Zurich (Brandli et al. 2014b; Mantel 2012)

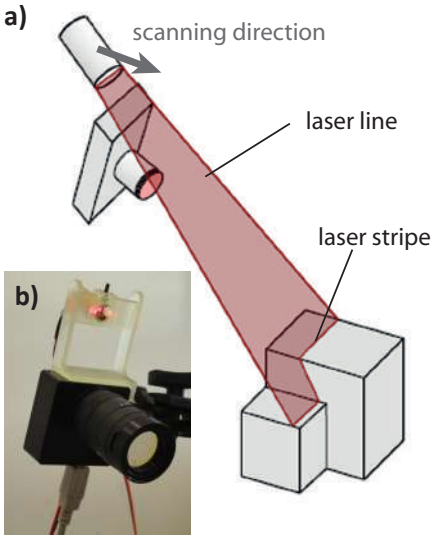


Figure 4.6: Setup used for the surface reconstruction using structured light. a) schematic representation of the setup with a scanned surface b) Photo of the DVS128 camera with optical filter and line laser rigidly mounted on top. Adapted from (Brandli et al. 2014b).

4.10.1 Setup

A simple but powerful light pattern that can be employed in structured lighting setups is a line (Forest et al. 2002). If the line is projected onto a surface and detected from a different inclination angle as in Fig.4.6a, the surface profile of the underlying structure can be extracted. If the system is moving relative to the surface, these profiles can be combined to reconstruct the full surface. If the laser stripe is extracted using a conventional image sensor, the sampling rate is restricted by the frame rate of the sensor which limits the resolution or the scanning speed of the system. By employing an event-based vision sensor, the sampling frequency can be set by the pulsing frequency of the line laser which allows the system to move with higher speeds.

In the experiments conducted in (Brandli et al. 2014b), a line laser was mounted on top of a DVS128 at an inclination angle of α_L of 8°. To improve the signal-to-noise ratio, the DVS was

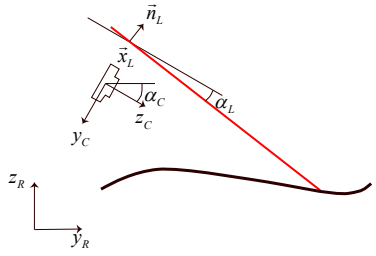


Figure 4.7: Coordinate system used in the surface reconstruction. Subscripts R is used for real world, C for camera and L for laser coordinates. Reprinted from (Brandli et al. 2014b).

equipped with an optical bandpass filter centered at the laser line wavelength of 636nm. The system was calibrated using a striped surface of well known distances. Furthermore it was assumed that the system can only translate and rotate in the plane spanned by the y- and z-axis.
To mark the onset of the laser pulses, specific laser trigger events Et were injected into the event stream using a dedicated input pin on the DVS128 camera. These events only carry a timestamp and a separate address index n. The address is only used to distinguish them from conventional events:

$$\begin{aligned} Et = &\{Ev : K(Ev) = triggermask \\ &\cap T(Et^n) < T(Et^{n+1})\} \end{aligned} \tag{4.14}$$

4.10.2 Laser Stripe Extraction

The key problem to be solved in this setup is to extract the laser stripe in the DVS output even in the presence of the events caused by the motion of the system. The used approach relies on the fact that the events triggered by the laser line are in sync with the laser pulse. Instead of simply filtering out all events outside of a fixed time window after the laser pulse, the developed algorithm measures the temporal delay distribution and uses it as a weighting function for the incoming events. The delay distribution is affected by multiple parameters such as bias settings, temperature or illumination of the scene and by constantly measuring this distribution, the algorithm can dynamically adapt to different settings. The weighting function deduced from this distribution allows assigning each incoming event a value which is proportional to the probability that it originates from a laser line pulse and thereby increases the signal-to-noise ratio.
To compute the nth delay histogram H_n, the event stream is sliced into separate event sets S_n^{ON} for ON and OFF events[2] using the laser trigger events:

$$\begin{aligned} S_n^{ON} = &\{Ev : T(Et_n) < T(Ev) < T(Et_{n+1}) \\ &\cap Pol(Ev) = 0\} \end{aligned} \tag{4.15}$$

These events are then binned into q temporal bins $D_n^{ON}(l)$ with index l and width fq where f is the pulsing frequency. To increase the stability of the histogram, the bins B_n of the histogram are averaged over the latest m laser pulses:

$$\begin{aligned} D_n^{ON}(l) = &\{Ev : Ev \in S_n^{ON} \\ &\cap \frac{l}{f \cdot q} \leq T(Ev) - T(Et^n) < \frac{l+1}{f \cdot q}\} \end{aligned} \tag{4.16}$$

$$B_n^{ON} = \sum_{j=n-m}^{n-1} \sum_{D_j^{ON}(l)} \|Sign(Ev)\| \tag{4.17}$$

$$H_n^{ON} = \{B_n^{ON}(l) : l \in [0, q-1]\} \tag{4.18}$$

This histogram is used to compute the scoring function P by normalizing the bin count with the total number of events T and subtracting the average bin count. This subtraction penalizes bins which are not correlated with the laser trigger events i.e. events that are caused by the uniformly distributed events from the system's motion and noise events (Fig.4.9).

$$T_n^{ON} = \sum \{B_n^{ON} : B_n^{ON} \in H_n^{ON}\} \tag{4.19}$$

$$P_n^{ON}(Ev) = \frac{\sum \{B_n^{ON} : Ev \in B_n^{ON}\} - \left(\frac{T_n^{ON}}{q}\right)}{T_n^{ON}} \tag{4.20}$$

For the laser stripe extraction each incoming event is multiplied with the scoring function and added to a score map $M_n(u, v)$ in the pixel coordinate space (u, v).

$$M_n^{ON}(u, v) = \sum_{S_n^{ON}} P_n^{ON}(Ev) + \sum_{S_n^{OFF}} P_n^{OFF}(Ev) \tag{4.21}$$

For the laser stripe extraction, the latest o score maps (o is usually around 3) are averaged

[2] The equations are only shown for the ON events but the OFF events are treated accordingly in separate sets and maps

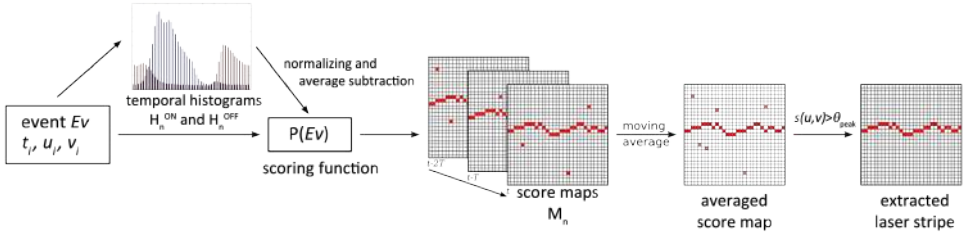


Figure 4.8: Overview over the laser stripe extraction algorithm. Incoming events are used to update the histogram and scoring function as well as to compute the score maps which are used to extract the laser stripe. Reprinted from (Brandli et al. 2014b).

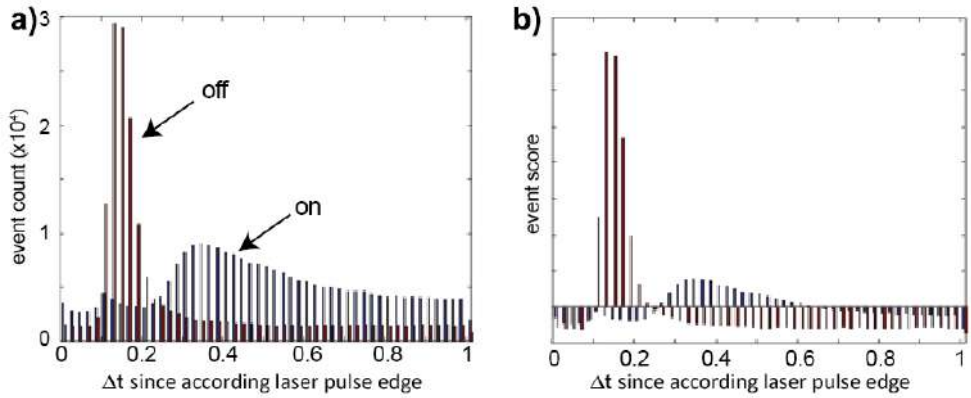


Figure 4.9: Example of event histogram and scoring function. a) measured delay histogram for ON and OFF events. b) Resulting scoring function. Reprinted from (Brandli et al. 2014b).

and the maximally scoring pixel per column which is above a threshold ϑ_{peak} (around 1.5) is considered to be part of the stripe. If multiple neighboring pixels are above the threshold, a weighted average is performed to determine the y position of the line.

4.10.3 Implementation

For an optimal performance of the algorithm, it had to be implemented in an event-based fashion as described in the following.

For the laser line extraction, multiple score maps have to be averaged. This can be done efficiently by accumulating data from o laser pulses and for each event checking whether the score is larger than the maximally scoring pixel. If this is the case the index of the according pixel and its value replace the old maximum. This allows to get the maximally scoring pixel at the end of the o pulses without a search.

To get sub-pixel resolution for the laser stripe extraction, the neighboring pixels on top and below within a column are searched one by one for further pixels with a score above the threshold and the search is aborted if a first pixel is below the threshold. The according score values are used to compute the weighted average over these pixels to get the y coordinate. This neighborhood search reduces the search space for pixels above the threshold significantly.

The m histograms are not continuously averaged as in a finite impulse response (FIR) filter. Instead if an event arrives it is directly added to

a preliminary histogram sum and after m pulses, this accumulated sum gets divided by m to replace the old histogram average.
These optimizations allow to reduce the computational load of the algorithm significantly.

4.10.4 Results

Figure 4.10: Scanned artificial landscape: in red the inclination of the laser and in blue the scanned area. From (Brandli et al. 2014b)

To qualitatively validate the approach, an irregular 3D surface was plotted and scanned with constant linear motion of about 1 m/s as shown in Fig.4.10. The according reconstruction results shown in Fig. 4.11 show that the approach delivers highly resolved surface information. Only for the parts of the surface not reached by the line laser, the error exceeds 5mm. A youtube video shows the algorithm and its output (*Adaptive filtering of DVS pulsed laser line response for terrain surface reconstruction* 2013).

4.10.5 Discussion & Outlook

The proposed approach measures the event delay statistics after a laser pulse by averaging multiple event delay histograms. These statistics are at the same time used as weighting criteria for extracting event signals which originate from the laser pulse. The approach produces quantitatively accurate results and would enable fast motions in unknown terrains if mounted on a robot.
The presented approach allows to gather depth information from multiple points in parallel which increases the measurement bandwidth compared to a single or multiple a point lidars. Structured light is not only suited for line patterns but can also be employed for more complicated patterns. A simple but powerful approach to produce a full depth map in a single pulse would be to use an electronic or mechanical shutter in front of a conventional Kinect to pulse the light pattern. With the event-based pattern detection, this depth map could be acquired at higher frequencies than with the existing Kinect or a Lidar. For the laser stripe extraction pulsing frequencies of 500Hz were achieved but experiments with a point laser showed that it might be possible to run the system at up to 2kHz if the light source is strong enough.

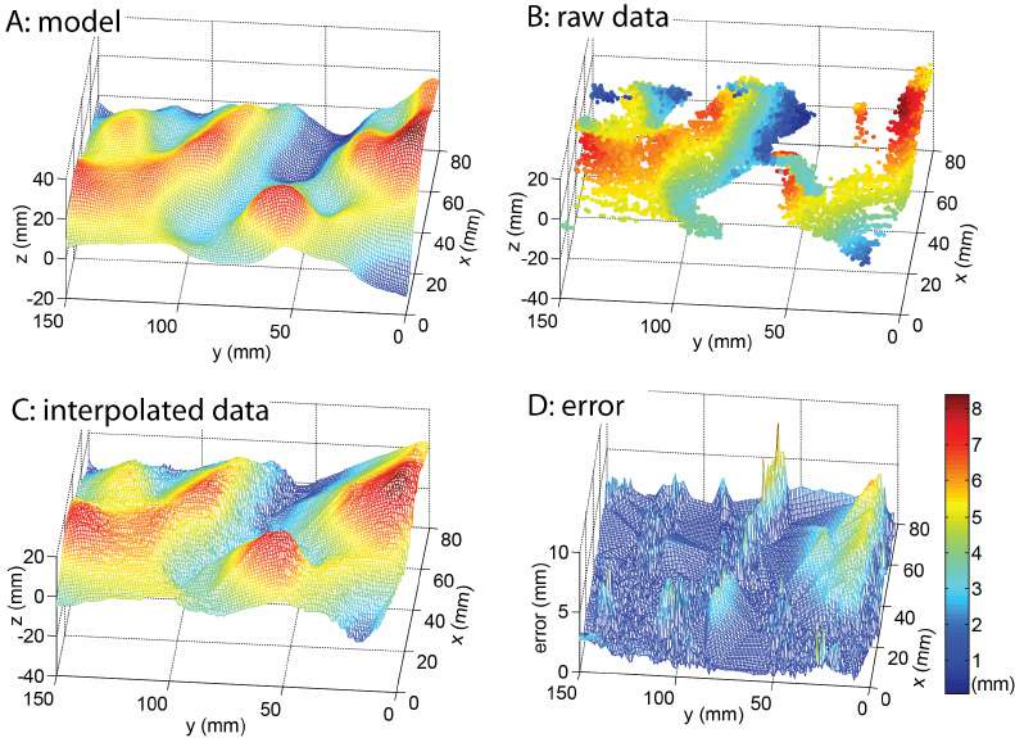


Figure 4.11: Surface reconstruction results. a) original data used to plot the landscape b) raw measurement points c) interpolated data using MATLABs TriScatteredInterp function d) Distance between closest reconstruction point and model aligned using iterative closest point (ICP). Reprinted from (Brandli et al. 2014b).

4.11 Study of DAVIS Decompression

The DVS inherently performs an in-pixel, asynchronous video compression (Fig.4.13). Knowing the absolute temporal contrast encoded by a DVS event would allow reconstructing the visual scene in real time and with sub-ms temporal resolution.
This work on such a real-time event decoding approach has been realized and co-developed in a collaboration with L. Muller(Brandli et al. 2014c).

4.11.1 Constant Decoding

The simplest way of decoding the DVS output is to assume that each event encodes a constant log intensity change $\theta_{ON} = -\theta_{OFF}$. Based on this assumption one can use a mechanical shutter and integrate the value counts per pixel. This approach was implemented in jAER as the "accumulate events" rendering option. Fig.4.12 shows the output of such a reconstruction and indicates its shortcomings:

1. To start the accumulation of the event values for all pixels at the same illumination level, the approach requires a mechanical shutter. This mechanical shutter is expensive and it also implies that many events have to be integrated if the pixels start in the dark and this is problematic as described in the next point.

2. Due to the mismatch across the pixels and the fact that the signal is discarded during the event communication and reset of the DVS

pixel, the temporal contrast across events is not matched. The assumption of a constant δ across all pixels leads thereby to an error which gets integrated.

3. The ON and OFF events can have different thresholds but if the same δ magnitude for both types is used for the decompression, it will drift towards the polarity with the lower threshold.

The frames from the DAVIS offer a solution to all three of these problems:

1. The frames can serve as starting point for the reconstruction and thereby replace the mechanical shutter.

2. By regularly acquiring frames, the integration of the error can be interrupted and a ground truth for the reconstruction can be acquired.

3. The actual θ_{ON} and θ_{OFF} can be measured so that the errors can be minimized and the drift reduced.

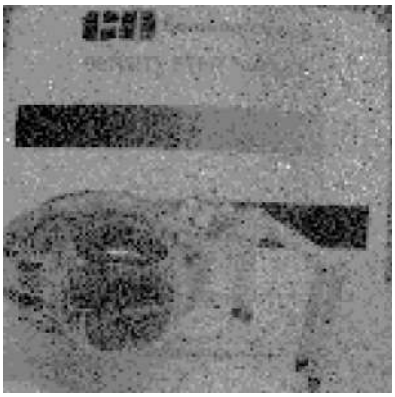


Figure 4.12: Decompressed DVS data of a toy spider in front of an Edmund density chart using the constant decoding approach implemented in the jAER "accumulate events"method (full scale black to white: 10 events). Reprinted from (Brandli et al. 2014b).

4.11.2 DAVIS Output Decompression

The DVS events of the DAVIS correspond to log intensity steps and therefore the first step upon the acquisition of an APS intensity sample image $\hat{I}_S$ is to perform a logarithmic compression $I_S = ln(\hat{I}_S)$. Incoming events are then used to update the brightness values on this decompressed image I_d: For each incoming ON or OFF event the according θ^i_{ON} or θ^k_{OFF} is added to the pixel value of the decompressed image I_d with index k.
θ^k_{ON} and θ^k_{OFF} are updated using a feedback controlled algorithm. For each incoming image, the difference between the latest reconstruction I_d and the acquired image I_S is computed to get a pixel-wise error signal: $E = I_d - I_S$. This error signal is then used as feedback signal, multiplied by a learning factor η_e and subtracted from the most recent θ estimation. If the θ was too large, then the I_d becomes too large and the error is negative so that the θ becomes smaller upon the next update. If a pixel has seen both OFF and ON events between two images, the error has to be attributed to the right polarity which is done through a multiplication of the error with the ratio of the events of this polarity over all events at a pixel with index k. The update for the $\theta_{ON,k}$ is given using following equation ($\theta_{OFF,k}$ is computed accordingly)

$$\theta^f_{ON,k} = \theta^{f-1}_{ON,k} - \eta_e \cdot E_k \cdot \frac{N_{ON,k}}{N_{tot,k}} \quad (4.22)$$

To accelerate the convergence of the θ estimates, in a first learning phase it is assumed that all $\theta_{ON,k}$ are similar and instead of learning the θ for each pixel individually, a global θ is learned. For stability this global estimate is low-pass-filtered using the rate η_a:

$$\theta^f_{ON} = \theta^{f-1}_{ON} \cdot (1-\eta_a) + \eta_a \cdot \sum_{k \in Pixels} \frac{\theta^f_{ON,k}}{N_{pixels}} \quad (4.23)$$

$$\theta^f_{ON,k} = \theta^f_{ON} \quad (4.24)$$

In addition to attributing the error to the right polarity, the performance of the algorithm

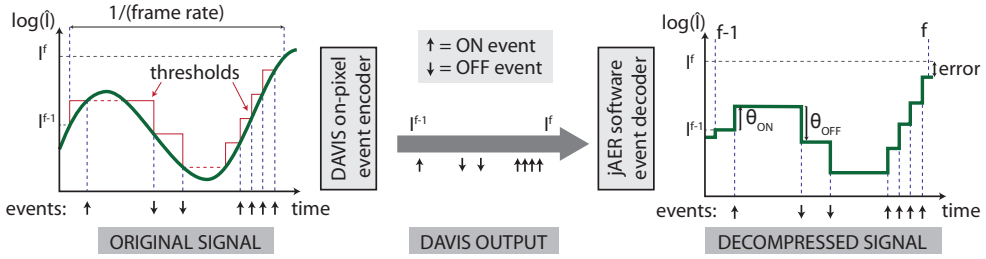


Figure 4.13: Decompression of the DAVIS output by decoding the θ intensities encoded by the events. Reprinted from (Brandli et al. 2014c).

was improved by binning the events into temporal bins representing the time since the last event arrived at a pixel. For each incoming event, the inter-event interval dt_{last} is computed by subtracting the last timestamp of given pixel from the timestamp of the event and according to this value, a θ_{ON} is chosen from one of 8 different values. These 8 temporal bins are spaced logarithmically because the underlying assumption is that if dt_{last} is small, there must be a lot of activity which increases the arbitration delay and thereby the value encoded by an event.

4.11.3 Results

4.11.3.1 Decompression Results

The results in Fig.4.14a-d shown how fast motions can be decompressed down to a temporal resolution of 1us. The decoding is implemented in jAER ("ApsFrameExtrapolartionISCAS") and runs in real-time on a Core-Duo i7 3.07GHz desktop computer. The traces behind the ball show the mismatch between the θ of ON and OFF events.
The results in Fig.4.14f show the quality of the reconstruction: it is good enough to see the steps and letters on the Edmund density step chart after a rotation and 1.2Mio events. The error in Fig.4.14h shows how most of the errors (abs(Fig.4.14f - Fig.4.14g)) are due to an imprint which is most likely because the first events are not generated from the reset level of the pixel which leads to wrong estimates.
For a quantitative analysis, a uniform white bar was swiped across the sensor's field of view at about 10cm/s. From this reconstruction of a uniform surface, the error ($abs(I_d - I_S)$) was measured and a result of 0.5DN (ADC steps) was achieved at 1.42ms APS exposure (white bar far from saturation level). For longer exposures such as 5.23ms (white bar just above saturation), the reconstruction error was about 2DN.

4.11.3.2 Compression ratio

The compression ratio CR indicates how much data was saved through a given processing step. In the case of the DAVIS it can be approximated using following formula:

$$\begin{aligned} CR &= \frac{UncompressedDataSize}{CompressedDataSize} \\ &= \frac{R \cdot S \cdot f_e}{R \cdot S \cdot f_s + eps \cdot b} \end{aligned} \tag{4.25}$$

where R is the intensity resolution in bits (10), S the size of the imaging array (240x180=43200), f_e the equivalent frame rate of a conventional imager, f_s the actual sampling rate of APS frames, eps the average event rate in Hz and b the bits required to communicate an event (32). The equivalent frame rate can be estimated as the inverse of the temporal precision i.e. jitter of the events which is around 1/500us so that f_e = 2kHz. The maximum compression ratio is achieved when nothing is moving and the sensor is producing only background events which leads with the sampling rate of 1Hz to CR = 1862. Under more realistic event rates (500keps) with a reduction of the equivalent frame rate to 200Hz

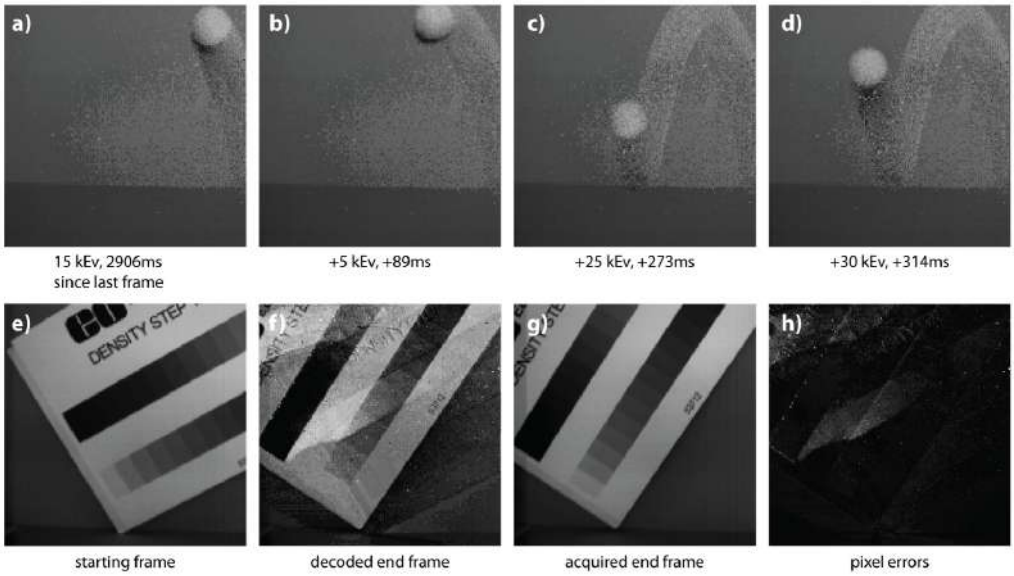


Figure 4.14: Decompressed DAVIS data. a)-d) Visual scene of a bouncing ping pong ball illustrating the high temporal resolution of the approach e) APS image of an Edmund density step chart (I_S^t) f) Decompression of the DAVIS data after the chart has been rotated (I_d) causing 1.2Mio events g) APS image of the rotated chart (I_S^{t+1}) h) reconstruction error which is used as feedback signal (E). Reprinted from (Brandli et al. 2014c).

(because higher temporal resolution might not be of interest), CR is still 5. For bigger sensors it is surmised that the compression ratio becomes bigger because the event rate grows linearly with the length of the contours whereas the size of the imaging array grows quadratically.

4.11.3.3 Discussion & Outlook

The proposed algorithm allows inferring the absolute light intensities and decompress the DAVIS sensor data in real-time. The feedback controller approach delivers promising results but it still needs further work to improve the quality of the decompression.
One of the shortcomings of the proposed approach is the fact that it lacks a strategy on how to handle events which are outside of the dynamic range of the APS images. Furthermore the timing of the image acquisition and the events has to be investigated in detail. Eventually the image might no longer be used to replace the reconstruction but only for the computation of the error signal. The reconstruction would also profit from using spatial information. If the reconstruction would be performed offline or with some delay, a non-linear approach could allow to have a temporally smooth reconstruction.

4.12 Study of Event-Based Line Segment Detector (ELiSeD)

For many applications it would be useful to have a general purpose purely event-based, low-level feature. The presented approach which was developed by C. Brandli and implemented by J. Strubel during his Master thesis, is such a low-level, event-based feature(Strubel 2013a).

4.12.1 The Event Correspondence Problem

Moving objects in the field of view of a DVS leave a continuous trace of events and if tracked properly, this continuous position estimation allows avoiding the spatio-temporal correspondence problem. For any feature in a frame at time t, such a tracker would allow to determine where it is in frame $t+1$ without the necessity to perform an extensive search and matching. But to allow this tracking, the "event correspondence problem", as termed from here on, has to be solved: For any two events it must be determined whether they have been caused by the motion of the same spatial feature. Each event has to be corresponded to a set of previous and following events caused by the same spatial feature.
A first approach to assign events to certain objects for tracking, is the so-called "RectangularClusterTracker" in jAER. Each incoming event is assigned to the closest cluster and if there is none within a certain radius, a new cluster gets created. While this method works fine to track individual moving objects, it struggles with bigger objects because it is not attached to a specific spatial feature.

4.12.2 Parametrized Spatial Descriptors

Under the optic flow assumption, one way to solve the event correspondence problem is to describe the spatial structures in a parametrized way. This approach was for instance used for the pencil balancer (Conradt et al. 2009b) where a continuous Hough tracker tracks position and angle of the pencil or in the jAER "Pig Tracker" to track a large number of pre-defined line segments. To generalize this approach, the parametrization must be capable of modeling any kind of spatial contrast. One possibility to do so is to model spatial contrast structures as a set of line segments so that curves are approximated as piecewise linear. Even though the Hough transform can also be used to detect line segments, the end point determination is a costly problem when executed event-based. The main problem is that for the end point determination, the algorithm has to iterate over all hypotheses affected by the addition or removal of an event and this is computationally expensive. For this reason a local, more bottom-up approach for line segment detection is desirable. The line segment detector (LSD) algorithm (Gioi et al. 2012) is such a bottom up detection algorithm and therefore it is an ideal candidate to be translated into an event-based low-level feature detector. The Event-Based Line Segment Detector (ELiSeD) described in the following is an event-based version of the LSD and a bottom-up line segment detector.

4.12.3 Line-Support Regions

Already in early releases of jAER, Delbruck's simple orientation filter was included. This filter assigns each event one of four discrete orientations by computing the orientation of the most recent activity in the pixels neighborhood. This can be done by storing the latest timestamp of an incoming event for all pixels ("last timestamps map", Fig.4.15b) and computing the the most coincident orientation on this timestamp map by taking the two dimensional derivative using Sobel edge detectors. This method does not necessarily have to be discretized as in the orientation filter and can also deliver a continuous orientation value for any event[3].
To translate the LSD into an event-based algorithm, an equivalence to the level line angle has to be determined. The orientation based on the derivative of the last timestamps map is an ideal

[3] which is what is done in ELiSeD

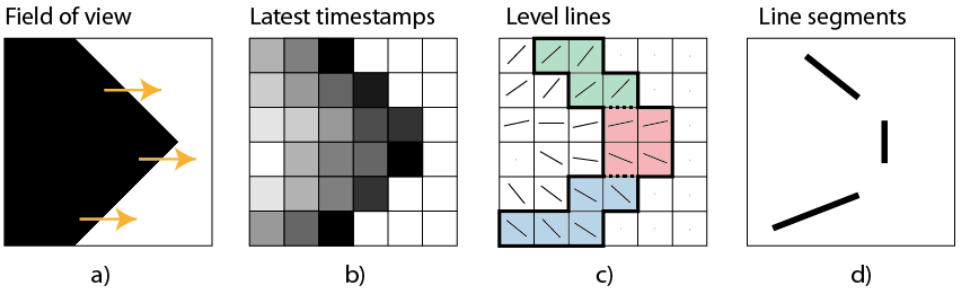


Figure 4.15: ELiSeD tracking a black corner moving right: a) visual input to the sensor, b) last timestamp map (dark = new, bright = old), c) level lines with differently colored support regions. uncolored pixels do not have events in the ring buffer and d) detected line segments.

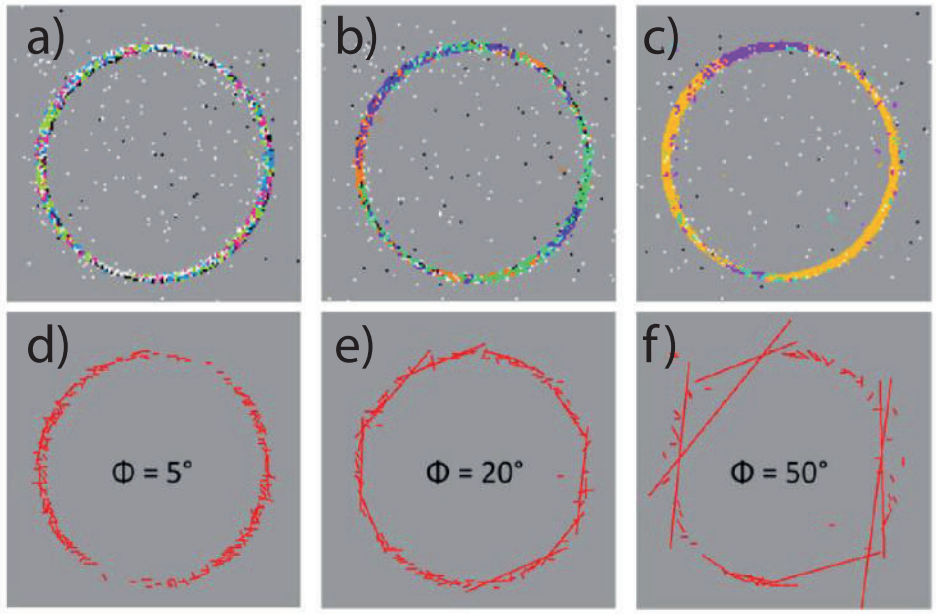


Figure 4.16: Line-support regions and line segments on a circle. a)-c) DVS events in black/white and in color different line-support regions for increasing tolerance angles ϕ = 5°, 20°, 50° d)-f) detected line segments. Reprinted from (Strubel 2013a).

candidate. 3x3 Sobel filters in x- and y-direction can be used to get the x- and y-components of the normal vector on the last timestamps map and the *arctan* to get the orientation angle (Fig.4.15c)).

The clustering of the events i.e. the creation of the line-support region is performed similar to the rectangular cluster tracker with the exception that an event can only be added to a support region if it has a similar orientation within a certain tolerance angle ϕ. The event buffering is done using a delta ring buffer. In case an event can be assigned to two different support regions, these regions are merged. The tolerance angle plays a crucial role for handling curvatures as shown in Fig.4.16. The coverage of how many events are

actually allocated to a support region/segment depends on the direction of the motion because motions in parallel with the spatial contrast don't cause sufficient events to be tracked as shown in Fig.4.17. Still the algorithm is capable of clustering all the relevant events of a line.

4.12.4 Line Fitting

While the original LSD is fits a rectangular bounding box over the support region, this would be too expensive to be performed for each event which is added or removed from or to a support region. Instead the line segment is approximated with the larger axis of the image moment ellipse of the support region (Fig.4.15d). The image moment is a weighted sum of the support region pixels which can be used to determine the center of mass. The second moment matrix of the support region can be used to calculate the eigenvalues and thereby the ellipse axes (Rocha et al. 2002). This computation can be performed event-based and it is cheap to add and remove events.

4.12.5 Results

Growing support regions from the event orientation can also segment curved shapes according to the tolerance angle as shown in Fig.4.16. The bigger the tolerance angle gets, the more events are merged into the same support region. Due to noise in the timing, the orientation of some pixels cannot be computed correctly which can disrupt the support regions.
Fig.4.17 shows how the support regions grow over the full field of view (Fig.4.17a based on the support regions shown in Fig.4.17b) and cover the edges of the shown chessboard pattern completely. But it also shows a shortcoming of the approach: if the sensor is moving along one axis (x or y), spatial contrast along the other axis cannot generate events and thereby not be described with line segments. This can also be quantified: For the diagonal motion of Fig.4.17b which contains x and y components, 65% of the events can be assigned to a line segment. The vertical motion of Fig.4.17d leads to less events along the horizontal direction and only 51% can be assigned.
Ideally a line segment should track a spatial edge feature for the full time it is in the field of view of the DVS. But most line segments actually catch on to small spatial structures and live much shorter as seen in Fig.4.18. A few short events live several hundered microseconds and a few long line segment live very short but most line segments can be found in the lower left corner (small support regions and short lifetime). This on one hand is a consequence of the highly complex scene observed: a moving leg in front of moving background but it also points out a shortcoming of the algorithm as it is. Stability and lifetime of the line segments needs to be improved.

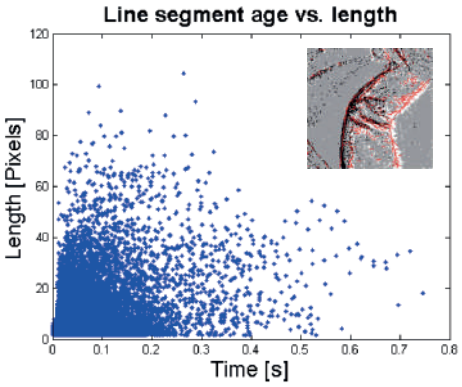


Figure 4.18: Lifetime vs. size of the ELiSeD line segments from a walking scene looking at leg (inset). Reprinted from (Strubel 2013a).

4.12.6 Discussion & Outlook

Even though the first implementation of the ELiSeD low level feature detector and tracker showed promising results, it still has to be improved. For straight lines, the algorithm sticks reliably with them but in more natural scenes, most line segments stay small and only live for a short time (Fig. 4.18).
Certain aspects of the algorithm still have to be explored and improved:

- The buffer size should be chosen to be pro-

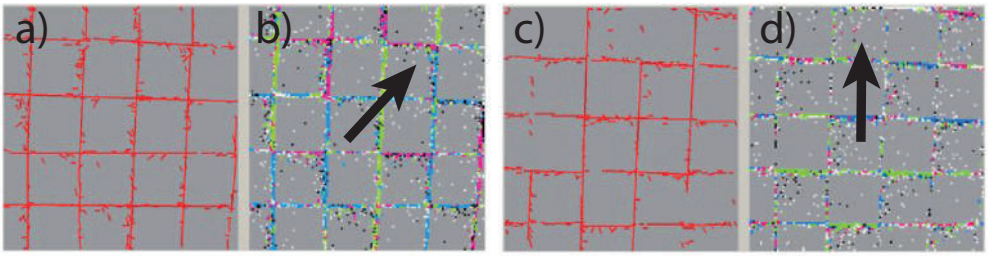


Figure 4.17: Coverage of the events with support regions dependent on stimulus motion. a) line segments and b) line-support regions for a digonal motion (black arrow): 65% of events can be attributed to a support region. c) line segments and d) line-support region for a vertical motion: 51% attribution. Reprinted from (Strubel 2013a).

portional to the total amount of spatial contrast in a scene. This could be done in two ways: either by estimating the total spatial contrast from the event distribution within the pixel array or by inferring it from the optical flow equation: If the event rate C^t is measured and the optical flow V inferred from the last timestamps map, the total spatial contrast can be estimated.

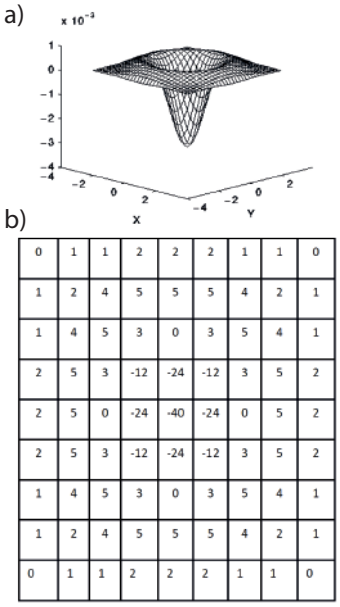
- The computation of the orientation could be improved by working with more events. This could be achieved by adding more events to the last timestamps map before the actual orientation is computed.
- A more sophisticated merging procedure might integrate mini-line segments into bigger ones and thereby make them more stable.
- Letting the line segments survive even without events depending on their size could make them more stable by bridging event-less episodes.
- A strategy to buffer events from line segments which are oriented perpendicularly to the motion has to be elaborated.

4.13 Study of Event-Based Keypoints

Under the optic flow assumption, the relative number of events should allow inferring spatial contrast without knowing the absolute light intensities. The local spatial contrast plays a crucial role in most keypoint detectors and descriptors. The work presented in this section is based on ideas of C. Brandli and implemented by Varad Gunjal during his Bachelor thesis (Gunjal 2012) and aims towards using the event distribution for fully event-based keypoint detectors and descriptors.

4.13.1 Pixel Buffer

To use conventional keypoint detectors and descriptors with the DVS output, a pseudo-image reflecting the local intensity distributions can be created. A simple way to do this is using a pixel buffer: each pixel contains a ring buffer and the number of ON events in the buffer is translated into the pixels intensity. The assumption behind this approach is that a bright pixel in a scene must have seen mostly ON events to become bright. By capturing the event history of a pixel, the brightness can be inferred. This pixel buffer was implemented with a delta ring buffer in which each ring buffer only stores the sum of ON events in the buffer and a binary buffer vector of the event history at this pixel: true = ON even, false = OFF event. This approach is similar to the accumulate events method where all events are added but with the advantage that old events (event that don't fit into the buffer anymore) get forgotten. This avoids an integration of the error and a drift since the maximal ON or OFF count is limited by the buffer size. Fig.4.19 shows the effect of increasing the buffer size: when the buffer is too small it cannot adequately represent the spatial contrast. When the buffer is too big, the buffer contains more entries than events are generated when moving from the darkest part of the scene to the brightest part. It can also be seen that the PixelBuffer approach suffers from the mismatch among the pixels. Another disadvantage is the fact that the pseudo-image depends on the motion direction.



0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

Figure 4.20: Convolution Kernel for the normalized Laplacian of Gaussian (sum of entries = 0). a) 3D plot of the kernel b) discretized version of the kernel. Reprinted from (Gunjal 2012).

4.13.2 Keypoint Detection Using Convolutions

To perform a SIFT-like keypoint detection, the pseudo image gets convolved with a discretized Laplacian of Gaussian kernel. This convolution is performed fully event-based. If a new event arrives at a pixel with the same polarity as the event removed from the ring buffer, then no update in the output is performed. If it increases (ON replacing OFF) or decreases (OFF replacing ON) the pixel value, then this delta is propagated through the convolution. A change in intensity affects all neighboring pixels whose convolution kernels include the pixel. Since the LoG is symmetrical (Fig.4.20) the delta can be multiplied with the entries in the kernel and added to the convolution output at the according pixel. Fig.4.21a shows the events as they come from the DVS128 sensor as well as the output of

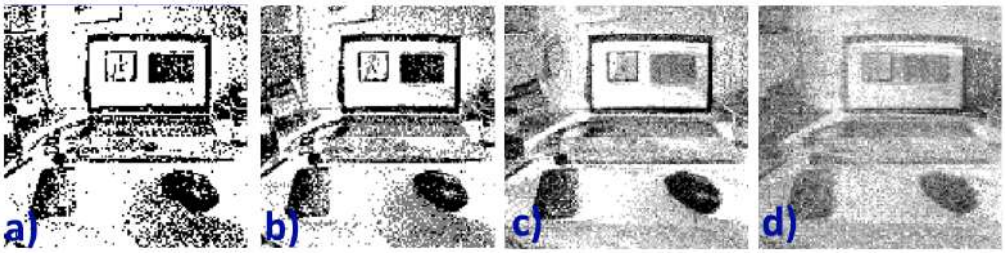


Figure 4.19: Output of the pixel buffer with different ring buffer sizes showing a scene with a desk and laptop. White = buffer filled with ON events and black = filled with OFF events. a) buffer size = 1 b) buffer size = 2 c) buffer size = 5 d) buffer size = 10.

the keypoint detection in the form or green pixels. These events are then used in a pixel buffer with depth 3 (4.21b) to generate a pseudo-image which is then convolved with the LoG: Fig.4.21c. The maxima in this convolution output which exceed a threshold then correspond to a blob i.e. a keypoint.

Table 4.1 Matching percentage of keypoints across scenes. **pattern from Fig4.22

Descriptor**	Same Scene	Different Scene
a)	100%	91%
c)	83%	8%
g)	91%	81%

4.13.3 Binary Keypoint Detection

By storing a binary string for each pixel which contains the data on which of the pixels on he FAST neighborhood ring Fig.4.3 are brighter and which ones are darker, FAST corners can be detected. The entries in these comparison strings can be updated as soon as a pixel in the pseudo-image changes value. This allows to detect new FAST corners instantaneously with a few comparisons.

4.13.4 Keypoint Description And Matching

To evaluate whether the proposed approach is partially functional, simple binary descriptor patterns were heuristically chosen (Fig.4.22a)-h)). To generate a keypoint description vector each of the pixels on the descriptor pattern around the keypoint pixel (denoted with X) is compared against the others in a pre-defined sequence. These N(N-1)/2 comparisons then make up the boolean keypoint description vector entries (e.g. if one pixel is brighter than the other, the according entry is true otherwise false). This vector is used to compare the similarity of a keypoint against another keypoint using the Hamming distance of their vectors (e.g. to solve the correspondence problem in stereo vision or object recognition).

To asses the quality of this approach, a random scene was recorded and its keypoints were compared to a another recording of the same scene as well as against a recording of a different random scene and the percentage of keypoints with a match in the other scene were noted. This matching is not very reliable since it is prune for false positive matches: with only 10 entries in the vector, it is very probable that two keypoints result in the same descriptor even though they are very different. This comparison also does also not assess how well the detector repetitively detects the same keypoints. It is not a all quantitative and was only used to see whether to recordings of the same scene generate more similar keypoint descriptors than another scene which seems to be the case:

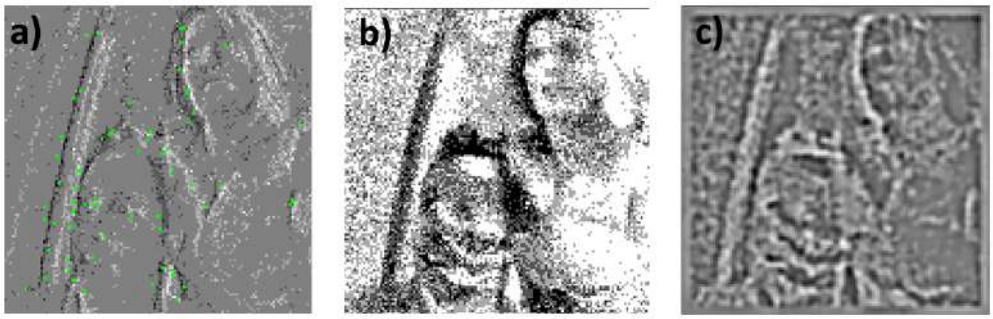


Figure 4.21: Feature detection in scene with two persons. a) DVS output. Keypoints are indicated with green points b) PixelBuffer pseudo-image with buffer depth 3 c) output of b) convolved with the LoG.

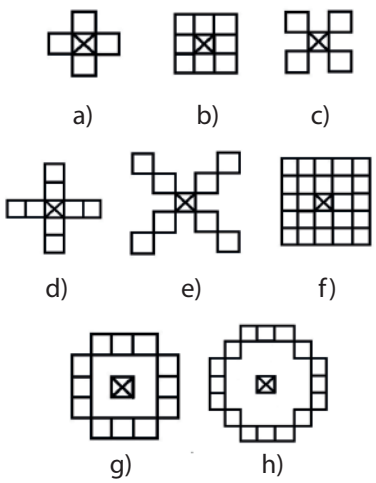


Figure 4.22: Shapes for different descriptor patterns used in (Gunjal 2012). The cross indicates the keypoint pixel. Reprinted from (Gunjal 2012).

4.13.5 Discussion & Outlook

The performance of the implemented approach is not yet optimized but the underlying idea of using the event statistics to infer the relative intensity distribution for an event-based keypoint detection and description could not be proven wrong. There are multiple ways to improve the presented approach:

- The pixel buffer performs poorly in reconstructing the observed scene. Using the event distribution to calibrate for mismatch would be a first step of improvement.
- The ON/OFF balance could be controlled using a feedback controller.
- The pseudo-image for the descriptor generation could be smoothed using a Gaussian kernel.
- Intensity reconstruction from a limited history of events is a crude method and more elaborate statistical measures could be used. Even a full intensity reconstruction with depth and camera motion estimation could be a possibility (similar to (Cook et al. 2011; Kim et al. 2014)).
- The SIFT keypoint rejection should be implemented to get rid of keypoints on edges.

Even though it might be possible to get this fully event-based approach to work, the APS images of the DAVIS can further facilitate the task.

5 Event-Based Machine Vision Applications

Principally event-based machine vision could be applied in all fields of machine vision but some applications are simpler to realize than others. Even more importantly some applications profit more from the event-based advantages than others.
To understand the most suitable applications, the unique selling propositions (USPs) of the technology have to be analyzed and matched with a market demand. For the DVS, which is the basis of most of today's EBMV, this looks like the following:

- *Low latency:* The low latency of the sensor make the sensor highly suited for real-time applications such as real-time interaction with the environment (e.g. robotics) or real-time display update (e.g. augmented reality).
- *Low power consumption:* The low power consumption make the sensors highly suited for battery powered applications such as mobile devices or mobile platforms as well as self-sustained sensing nodes or solar-powered devices.
- *Wide dynamic range:* The wide dynamic range of the sensors is useful in outdoor applications or uncontrolled lighting.
- *Sparse Data:* The sparse data removes the requirement for background subtraction which allows to simplify for instance tracking tasks.
- *Anonymous data format:* The DVS data does not allow to clearly identify people and is therefore raising less privacy and security concerns. This makes it highly useful for the application in private environments (households).

In the following existing as well as possible applications are discussed to highlight the pros and cons of event-based machine vision systems.

5.1 Robotics

One of the fields that uses machine vision already extensively is robotics. Especially autonomous robots can profit from EBMV because they are battery-powered, real-time, outdoor systems.

5.1.1 Robotic Arms

In robotic setups in which a robotic arm is fixed, the DVS facilitates controlling the robot by tracking the arm as well as the manipulated object and thereby allows to close the control loop with a low latency.
One of the first robots that was built to demonstrate the low reaction time of the sensor and sensing pipeling was the *RoboGoalie* which is capable of deflecting ping pong balls shot at a goal by moving an arm sideways (1 DOF) (Delbruck et al. 2007; Delbruck et al. 2013).
Another demonstrator that shows how low sensor latency simplifies control problems is the so called *pencil balancer* which is capable of balancing a pencil placed upright on a platform actuated by two arms moving in a plane (2 DOF). Two perpendicularly mounted DVS are tracking the position and the inclination angle of the pencil using a simplified Hough tracker to then counterbalance the fall motion by moving the platform in the opposite direction (Conradt et al. 2009a; Conradt et al. 2009b; Conradt et al. 2009c).
In another application the DVS is used to track a microrobotic arm through a microscope and deliver haptic feedback to the operator (Bolopion et al. 2012; Ni et al. 2012).

5.1.2 Humanoids

Due to the pre-processing performed on the DVS, it is easy to implement simple robotic tasks with this sensor. Simple controllers can even be implemented on FPGAs and used to control toy humanoid robots (Linares-Barranco et al. 2007).
DVS have also been integrated into more advanced platforms such as the iCub to solve more complex tasks such as feature based object recognition (Bartolozzi et al. 2011a; Bartolozzi et al. 2011b; Wiesmann et al. 2012) or to evaluate attention models (Rea et al. 2013).

5.1.3 Ground Robots

By suppressing background information, the DVS allows to simplify the line/lane detection and tracking which can be used to implement fast line follower robots (Brandli 2008; Ritz 2008).
By interfacing the DVS to artificial spiking neuron platforms such as SpiNNaker bio-inspired sensory-motor control loops can be implemented on simple ground robots (Denk et al. 2013).
The DVS can deliver information on the environment and terrain in which a legged robot is moving. This information can be acquired using a learning algorithm (Monteiro 2013) or surface reconstruction using a pulsed laser and a DVS (Brandli et al. 2014b).
In another scenario, an ATIS was mounted on a pioneer robot and the events are used to compute the time-to-contact for obstacle avoidance based on visual flow (Clady et al. 2014).

5.1.4 Unmanned Aerial Vehicles

For unmanned aerial vehicles (UAVs) and especially for micro-aerial vehicles (MAVs), latency is critical for a stable control and maneuvers in cluttered and dynamic environments.
With the goal to replace expensive, passive marker motion tracking systems that are often used for MAV control during aggressive maneuvers, a MAV was equipped with blinking light emitting diodes (LEDs) which are tracked by a DVS (Censi et al. 2013).
With the help of event-based Hough trackers, the pose of quadrotors can be tracked even during aggressive maneuvers (Mueggler et al. 2014).
The output of the DVS can also be used for optomotor heading regulation that can for instance be applied in UAVs (Censi 2015)

5.2 Automotive

The ATIS has been used in cars for pre-crash control and side impact airbag control (Schoitsch et al. 2013).
The DVS has can be used to analyze the surface in front of vehicles (esp. useful in motorcycles) using a Fourier transform and with the help of polarity filters wet streets can be detected (Dankers 2014). The DVS has been applied to track the hands of a driver during the driving task (Ríos et al. 2014).

5.3 Surveillance

Traffic surveillance has extensively been done by the group at AIT (Litzenberger et al. 2006a; Litzenberger et al. 2006b; Litzenberger et al. 2007b; Bauer et al. 2007; Belbachir et al. 2007) as well as pedestrian surveillance (Belbachir et al. 2010b; Schraml et al. 2010a).

5.4 Healthcare

Already before the creation of event-based vision sensors there have been efforts to integrate neuromorphic vision sensors into prostheses (Chiang et al. 2004). The most promising progress towards using event-based vision sensors for prosthetic devices is pursued by the French company Pixium Vision (Picaud et al. 2014).
Instead of interfacing the DVS information directly with the neurons, it can also be used for tactile displays (Ros et al. 2011).
Apart from prosthetics, the DVS technology can also be applied for fall detection in elderly care (Fu et al. 2008; Belbachir et al. 2011b; Belbachir et al. 2012a).

5.5 Industrial

Due to its high temporal resolution, the DVS can be applied for industrial inspection (Belbachir et al. 2010a) where the line sensor data can be translated into frames for further inspection (Hofstatter et al. 2009).
The temporal resolution for the application of event-based sensors in industrial tasks was investigated (Perez-Pena et al. 2011).
Another application is real-time particle tracking to control flows (Bömmels et al. 2004; Drazen et al. 2011; Borer 2014).

5.6 Entertainment

Since the DVS can be used for gesture recognition it was proposed as touch-less gesture user interface (Lee et al. 2012a). It was also investigated whether the DVS can be used for motion analysis in sports (Litzenberger et al. 2012).

6 Conclusion and Outlook

This chapter presents a conclusion to the thesis, a summary of the field of event-based machine vision, an outline of the author's contributions and an outlook on potential future directions of the field.

6.1 Conclusion

Neural computation differs fundamentally from symbolic computation not only on a conceptual level but also in its implementation. Nervous systems use analog, asynchronous highly parallel circuits which outperform the digital, synchronous and serial symbolic computing machines in various aspects. The field of neuromorphic engineering aims to replicate neural computation in silicon and developed a multitude of sensors. While many of these sensors struggled with transistor mismatch, the dynamic vision sensor (DVS) copes comparably well with this problem. The resulting high dynamic range, low latency and low power consumption are highly suited for a multitude of applications. First algorithms were developed for the DVS already shortly after its creation and it was shown that especially in tracking tasks, the sensor performs well.
The lack of access to absolute light intensities is a drawback for classification and recognition tasks using the DVS. The dynamic and active pixel vision sensor (DAVIS) which was co-developed in this thesis solves this problem with a second completely independent readout. By adding only 5 transistors to the pixel circuit, global shutter gray-scale images can be acquired without interfering with the DVS output. While conventional cameras were originally developed to produce beautiful pictures and movies for human interpreters, the DAVIS is a sensor optimized for computational vision. The DAVIS is a major advancement for the DVS technology not only because of its new readout modality but also because the DVS specifications were improved. The power consumption was lowered to an average of 10mW, the latency was reduced down to 3us and the dynamic range increased to 130dB. While an increasing number of algorithms and hardware around the DVS were developed in the past years, a common umbrella for this research was missing. Today more and more researchers from outside the neuromorphic community work with the DVS. These researchers are not interested in building bio-inspired systems but in developing better solutions for existing machine vision problems. The field definition given in this thesis can unite the research and accelerate it with the given review of the existing work. The discussions on the implementation of event-based vision algorithms are a further step towards a more formalized way of solving problems with event-based vision. The delta ring buffer presented in this thesis is a powerful tool and has been used in multiple of the presented algorithms.
The algorithms developed in collaborations and presented in this thesis tackle basic event-based machine vision problems such as event-based structured lighting, event decompression or event-based low-level features. The surface reconstruction algorithm using a line laser is the first event-based structured light algorithm and it allows surface reconstructions at pulsing frequencies up to 500Hz. The decompression algorithm is the first algorithm for event decompression using online error feedback. For a success of the field it will not only be critical to find niche applications for which the sensor is highly suited but to develop an general way of handling the data on an abstract level. The field would could experience a substantial growth if robust low-level features (equivalent to SIFT in frame-based machine vision) are developed. The developed event-based keypoint detectors and descriptors delivered useful insights into the problem and the event-based line segment detector (ELiSeD) shows promising results.

6.2 Development of Event-Based Machine Vision

Even though some of the existing event-based sensors and event-based vision algorithms might have been created with other motivations than the one stated in the field definition on page 28, the amount of work that can be considered being part of event-based machine vision has reached a respectable size in the last years. These are some of the corner stones in the development of event-based machine vision:

- Most of the developments in the field can be attributed to the creation of the DVS (Lichtsteiner et al. 2008) which was designed to cope with transistor mismatch and thereby produce reliable output data. The commercial availability of the sensor through iniLabs as well as the development of the jAER framework which allows an easy interaction with the sensor further accelerated the development of algorithms.

- The group around the DVS inventor T. Delbruck improved the sensor technology for instance by designing color sensitive pixels (Berner et al. 2011) and new biasing circuits (Yang et al. 2012) as well as a new generation of DVS sensors capable or capturing absolute intensities, the DAVIS (Brandli et al. 2014a). Multiple demonstrators to highlight the advantages of the technology have been developed such as the pencil balancer (Conradt et al. 2009c) and the robo goalie (Delbruck et al. 2013). Algorithms for particle tracking have been developed (Drazen et al. 2011) as well as for fall detection (Fu et al. 2008).

- The work at the Austrian Institute of Technology (AIT) not only resulted in new types of DVS-based sensors such as the ATIS (Posch et al. 2011b) or the 360° DVS line scanner (Belbachir et al. 2012b) but also first commercial applications have been developed. The DVS technology is used in the field of traffic surveillance (Litzenberger et al. 2007b), pedestrian detection and tracking (Belbachir et al. 2010b), industrial machine vision (Belbachir et al. 2011c) and panoramic photography (Belbachir et al. 2014).

- The research by R. Benosman et al. lead to theoretical insights and new algorithms for the DVS data. The research in the field of visual flow (Benosman et al. 2014) and stereo vision (Carneiro et al. 2013) formalized existing approaches in a mathematical way and produced new insights.

- J. Conradt et al. miniaturized the sensor for robotic applications and developed LED trackers (Muller et al. 2011) as well as first 2D SLAM algorithms (Weikersdorfer et al. 2013).

- B. Linares-Barranco and T. Serrano-Goterrando developed more sensitive DVSs (Serrano-Gotarredona et al. 2013b) and dedicated hardware for event-based processing such as convolution chips (Camunas-Mesa et al. 2012).

- P. Hafliger et al. have developed time-to-first-event imagers (Olsson et al. 2008b; Olsson et al. 2008a) and dual mode sensors (Lenero-Bardallo et al. 2014).

- D. Scaramuzza and A. Censi are using the DVS technology in the field of micro-aerial vehicles and have developed algorithms for visual odometry (Censi et al. 2014), pose estimation (Mueggler et al. 2014) and heading regulation(Censi 2015).

- J. Lee and colleagues are developing gesture recognition systems using the DVS (Lee et al. 2014).

- The company Pixium Vision is using event-based technology in their visual prostheses for blind people.

- The group of P. Dudek has taken a different approach to event-based machine vision. Instead of just computing temporal contrast, their pixels integrate a reduced instruction set processor which allows to perform more complicated computations in the pixel (Carey et al. 2013).

The field is developing with increasing speed and the first commercial applications will soon be available on the market which will further accelerate the growth.

6.3 Contributions to the Field

I started my thesis with the motivation to do event-based machine vision from the transistor up to the application. During my thesis I contributed to following aspects of the field.

6.3.1 Field Definition and Review

This thesis contains not only a description of the field, but it is also the most recent and most extensive review of event-based vision sensors, algorithms and applications. I came up with a field definition, contextualized the field and structured the existing findings. I also developed a new event notation which is more general than some of the ones used in the past.

6.3.2 DAVIS Sensors

Most of my work was focused on chip design which is why I tackled the algorithm development in collaborations. For the development of the DAVIS chip designs I did the following:

- designed[1] the delay line elements for SeeBetter10/11 to test the effects of bigger chips on the AER handshake protocol.
- reorganized the existing Tanner chip design cells into libraries for a more efficient design flow.
- designed the first DAVIS pixel (according to R. Berners idea & concept) for the DAVIS 64a.
- developed and designed the APS readout on the DAVIS 64a chip.
- developed the initial firm- and software for the APS readout of the DAVIS chips.
- helped moving the DAVIS designs from Tanner to Cadence for the design of DAVIS 240a.
- developed a Skill script to generate the schematics and layout for the AER arbiter, encoder and logic. This script was used in all Cadence DAVIS designs as well as the most recent cochlea designs and (Berner et al. 2014).
- co-designed DAVIS 240a with R. Berner (each about 45% of the work): he did the pixel and I did the pad frame, the AER arbiters, readout and interfaces which we then assembled together.
- did the APS characterization for DAVIS 240a.
- wrote the Skill scripts to move DAVIS 240a to the Tower process in which DAVIS 240b (chip design work split similar to DAVIS 240a) was produced, developed a global shutter readout and did the power routing of the chip.
- characterized DAVIS 240b except for the contrast sensitivity and DVS FPN. I documented the testing protocols and st up a test bench (black box) for the according measurements.
- developed multiple naming conventions, Cadence libaries, Skill scripts and documentation standards which allow to accelerate the chip design.
- came up with the arrangement for the pixels in the DAVIS RGBa.
- completely assembled DAVIS 240c, DAVIS 346a, DAVIS 346b and DAVIS 640a.
- assembled DAVIS BSIa except for the pads.

6.3.2.1 jAER

To handle the DAVIS chips in jAER, I developed the according chip and interface classes. I restructured the way events are rendered to be more efficient. I rearranged the event data format and iterating loops to handle the new APS events.

[1] Schematics and Layout

6.3.3 Event-Based Vision Algorithms

Many of the ideas for algorithms and software I had in mind, I couldn't implement myself because of a lack of time which is why I realized them in student projects and collaborations. I have supervised the following student projects:

- **Thomas Mantel, Spring 2012**: Semester Thesis "Application of a novel dynamic vision sensor for surface reconstruction" (Mantel 2012; Brandli et al. 2014b) supervised with M. Hutter and M. Höpflinger (ASL, ETHZ), co-supervised by Prof. T. Delbruck (INI, ETHZ) and R. Siegwart (ASL, ETHZ).
- **Markus Thurnherr, Spring 2012**: Bachelor Thesis "Prosthesis With Eyes - A Feasibility Study" (Thurnherr 2012) supervised with A. Pagel and S. Pfeifer (SMS, ETHZ), co-supervised by Prof. T. Delbruck (INI, ETHZ) and R. Riener (SMS, ETHZ).
- **Varad Gunjal, Fall 2012**: Semester Thesis "Development of Feature Descriptors for Event-Based Vision Sensors" (Gunjal 2012) main supervision, co-supervised by Prof. T. Delbruck (INI, ETHZ).
- **Jonas Strubel, Fall 2012**: Semester Thesis "Quadrotor tracking and pose estimation using a dynamic vision sensor" (Strubel 2013b; Censi et al. 2013) supervised with A. Censi, co-supervised by Prof. T. Delbruck (INI, ETHZ) and D. Scaramuzza (RPG, UZH).
- **Jonas Strubel, Fall 2013**: Master Thesis "Knee Tracking for Control of Active Exoprostheses using a Dynamic Vision Sensor" (Strubel 2013a) supervised with A. Pagel (SMS, ETHZ), co-supervised by Prof. T. Delbruck (INI, ETHZ) and D. Scaramuzza (RPG, UZH).
- **Luca Longinotti, Fall 2013**: Bachelor Thesis "A framework for event-based processing on embedded systems" (Longinotti 2014) main supervision, co-supervised by Prof. T. Delbruck (INI, ETHZ).
- **Susanne Keller, Fall 2014**: Bachelor Thesis "Title to be determined (ELiSeD)" main supervision, co-supervised by Prof. T. Delbruck (INI, ETHZ).

6.3.3.1 cAER

By proposing and conceptualizing the idea of a C-based event processing framework as well as by supervising and advising the development and implementation of cAER by Luca Longinotti (Longinotti 2014), I initialized the creation of a platform for embedded event-based machine vision applications. This work has the potential to grow much bigger in the future because the event-based approach has due to its efficiency a lot of potential in embedded applications and Luca did a great job.

6.3.3.2 Structured Lighting Using Event-Based Sensor

I have co-supervised the student project of T. Mantel on surface reconstruction using a pulsed laser line and a DVS sensor (Mantel 2012; Brandli et al. 2014b; Brandli et al. 2012). I suggested learning a temporal weighting function relative to the signal onset for the purpose of detecting pixels with events from the light source. For the publication of this work, I formalized the approach and optimized its performance. This approach has to potential to be used in high-speed structured lighting approaches and for this reason it has been filed for a patent. Unfortunately the surface reconstruction has not yet been used in its intended application: step planning for a quadruped robot.

6.3.3.3 DAVIS Decompression

In a collaboration with L. Muller (work share 50/50) we have implemented a real-time algorithm which allows decompressing the DAVIS data to create a video with high temporal resolution (Brandli et al. 2014c). This work can not only be used for video reconstruction but it also allows to assign events a quantitative value of temporal contrast which can be useful in a multitude of algorithms.

6.3.3.4 Event-Based Keypoints

The Bachelor thesis of V. Gunjal offered the possibility to realize an algorithm idea I had: Detecting conventional keypoints and creating conventional keypoint descriptors from local, relative intensity reconstructions (Gunjal 2012). Even though this work could not deliver convincing results in the limited amount of time of a BSc. thesis, the approach seems to be worth further investigation and might be combined with the DAVIS decompression approach.

6.3.3.5 Event-Based Line Segment Detector (ELiSeD)

During the Master thesis of J. Strubel on leg tracking which I co-supervised, I used the opportunity of working with a skilled programmer to realize an idea I was working on: the event-based line segment detector (ELiSeD) (Strubel 2013a). The promising results on this low level event-based feature lead to an ongoing continuation of the project (BSc. thesis of S. Keller) where the features will be used in a structure from motion setup. The ELiSeD and other low level event-based features that solve the event correspondence problem have the potential to become one the fundamentals in event-based machine vision.

6.3.3.6 Delta Ring Buffer

The event buffering principle which leads to an efficient, velocity-independent way of processing event-based visual information was developed by myself and used multiple of the stated algorithms.

6.3.3.7 Event-Based MAV Tracking

In a collaboration with D. Scaramuzza and A. Censi, the student J. Strubel realized a LED tracker to estimate the pose of a micro-aerial vehicle (MAV) (Censi et al. 2013). The goal behind this project was to replace the expensive tracking systems used in most MAV controlling setups. Apart from supervision and advise I only marginally contributed to this project.

6.3.3.8 Event-Based Leg Tracking for Prosthesis

In a collaboration with the Sensory-Motor Systems Lab (SMS) at the ETH, a feasibility study for using the DVS on a active knee prosthesis to track the sound leg for prosthesis control was performed by M. Thurnherr (Thurnherr 2012). Since this feasibility study came to the conclusion that it should be possible to track the leg, J. Strubel then implemented algorithms to track the leg under the co-supervision of the SMS, the Robotics and Perception Group (RPG) and the INI Sensors Group (Strubel 2013a). Even though I invested a considerable amount of time into to supervision of these two theses, the only tangible outcome was a first implementation of the ELiSeD. The main reason for this is the complexity of the problem which we initially completely underestimated: In a high-contrast scenario with a moving camera, it is hard to find a robust classifier that can distinguish events from a leg from events from the background.

6.3.3.9 Event-Based Keypoint Tracker

This unpublished but patented work which is not covered in this thesis was developed in a collaboration with M. Osswald. It might be a potential interface between event-based and conventional machine vision("Method for tracking keypoints in a scene").

6.4 Outlook

For a substantial and deep impact of EBMV in the field of machine vision, there are several challenges to be tackled which are described in the following.

6.4.1 High Quality Event-Based Vision Sensors

The sensors that have been developed and published so far have not yet been fully optimized for performance and served more as proof of principle. They were not designed as actual products to serve a specific market. There are efforts within several companies to develop and

produce event-based vision products but at this point the characterization of these chips has not been published.
In the following certain aspects of the sensors design that should be improved are discussed:

6.4.1.1 Resolution and Size

At the moment the ATIS is the event-based vision sensors with the highest available resolution of 304x240. But this resolution is only possible because the chip is very large: 9.9x8.2mm^2 which requires at least 2/3" heavy optics. The new VGA DAVIS will have a pixel resolution of 640x480 on an area of 13.5x10.mm^2 and even though it can be assumed that this resolution is sufficient for most machine vision applications, the chip will be too big for embedded systems. One way to increase the resolution is by moving to a smaller process node and thereby shrinking the pixel. In a 90nm BSI process the DVS pixel was be realized with a pitch of 12um (Berner et al. 2014) (the current DAVIS designs are 18.5um). Another approach to decrease the pixel size would be to use stacked image sensors with the photodiode on a different wafer than the DVS circuitry. This was actually the original goal behind the SeeBetter wafer but it is not clear if the required density and reliability of the through silicon vias (TSV) will ever become commercial.

6.4.1.2 Power Consumption

The power consumption of the DAVIS can significantly be improved with on-chip ADCs such as the ones that were added to the upcoming DAVIS designs (work by R. Berner et al.). But also the digital parts of the chip especially the digital pads can be optimized for power consumption which would have a significant impact because they consume most of the power. First steps towards this direction were already taken in (Berner et al. 2014).

6.4.1.3 Readout

For many machine vision applications, the AER data is interfaced to clocked processing architectures and it would be more efficient to use a semi-asynchronous readout scheme without the requirement of a full four phase handshake. The events could be treated similar to flags or interrupts in conventional processing architectures.

6.4.2 Optimized Processing Hardware

The development of optimized processing hardware and interfaces for events and event-based computation would allow to propagate the advantages of events through the full processing pipeline. One possibility for optimized hardware would be asynchronous CPUs and FPGAs which preserve the low latency and low power consumption of the data.

6.4.3 Performance Metrics and Benchmarks

To make event-based machine vision a real engineering discipline its output has to be quantified. For such a quantification, performance metrics and benchmarks have to be developed. Standardized benchmarks and datasets would allow to compare different EBMV algorithms against each other.

6.4.4 Low-Level Features

One of the success factors of machine vision was the development of SIFT, a universal "language" to represent the content of images in a compact way that is independent of rotation, scale, translation or brightness. If such a low level feature could be developed for the event-based data, most machine vision problems could be solved in a similar but more efficient way than frame-based approaches. The proposed event-based keypoint descriptors or the ELiSeD features are a first step towards this direction but they require further work.

6.4.4.1 Standardization

To facilitate the integration of new sensors and algorithms, certain standards such as communication and processing protocols should be developed (first examples can be found in (Serrano-

Gotarredona et al. 2009) or (Liu et al. 2014)). This standardization also facilitates collaborations and accelerates the development and integration of new event-based vision technology because work can be reused amongst research groups. The valuable work of inilabs standardizing the event format across multiple sensors should be continued and extended.

6.4.4.2 Libraries

Similar to OpenCV, public, well-documented code libraries should be created. jAER and cAER is a first steps towards this direction but the existing code should be cleaned up, better documented and separated into dedicated libraries. cAER has a great potential in this regard due to its modular structure.

6.4.4.3 Development Routines

The most efficient way of developing EBMV algorithms and software should be investigated and documented for teaching purposes. In addition it might be useful to develop a dedicated programming language or coding style. jAER offers an easy access to do event-based computation but good practices or formalized development routines are missing.

A Appendix

A.1 Publications

Berner, R., C. Brandli, M. Yang, S.-C. Liu, and T. Delbruck (2013a). "A 240 x 180 120dB 10mW 12us-Latency Sparse Output Vision Sensor for Mobile Applications". English. In: *2013 International Image Sensors Workshop (IISW)*. Snowbird, UT, USA. URL: http://www.imagesensors.org/Past%20Workshops/2013%20Workshop/2013%20Papers/03-1_005-delbruck_paper.pdf (visited on 09/24/2013).

Berner, R., C. Brandli, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck (2013b). "A 240x180 10mW 12us Latency Sparse-Output Vision Sensor for Mobile Applications". In: *2013 Symposium on VLSI Circuits (VLSIC)*. Kyoto Japan, pp. C186–C187.

Brandli, C., R. Berner, M. Yang, S.-C. Liu, and T. Delbruck (2014a). "A 240 x 180 130 dB 3 us Latency Global Shutter Spatiotemporal Vision Sensor". In: *IEEE Journal of Solid-State Circuits* Early Access Online.

Brandli, C., L. Muller, and T. Delbruck (2014). "Real-Time, High-Speed Video Decompression Using a Frame- and Event-Based DAVIS Sensor". In: *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 686–689.

Brandli, C., M. Osswald, and T. Delbruck. "Method for tracking keypoints in a scene". Pat. EP14178447.0 (pending).

Brandli, C., T. Delbruck, M. Hpflinger, M. Hutter, and T. Mantel. "Method for reconstructing a surface using spatially structured light and a dynamic vision sensor". Pat. EP14178447.0 (pending).

Brandli, C., T. Mantel, M. Hutter, M. Hpflinger, R. Berner, R. Siegwart, and T. Delbruck (2014b). "Adaptive Pulsed Laser Line Extraction for Terrain Reconstruction using a Dynamic Vision Sensor". In: *Frontiers in Neuromorphic Engineering* 7, p. 275. URL: http://www.frontiersin.org/Journal/10.3389/fnins.2013.00275/abstract (visited on 01/04/2014).

Censi, A., J. Strubel, C. Brandli, T. Delbruck, and D. Scaramuzza (2013). "Low-latency localization by active LED markers tracking using a dynamic vision sensor". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 891–898.

A.2 Curriculum Vitae

Christian Brändli

Curriculum vitae

CONTACT INFORMATION

Christian Brändli
St. Georgstr. 6
5400 Baden-CH

Work: (0041) 44 635 30 61
Mobile: (0041) 77 430 32 59
E-mail: ch.braendli@gmail.com

CITIZENSHIP

Switzerland, born 2-Feb-1985
Place of origin: Rorbas ZH

RESEARCH INTERESTS

Event-Based Machine Vision, Computer Vision, Neuromorphic Engineering, Bio-inspired Robotics, Computational Neuroscience, Mixed Signal Design

EDUCATION

ETH Zurich

PhD student, Institute of Neuroinformatics **Feb 2011 - Jan 2015**

- Supervisor: Prof. Tobi Delbruck (Sensors Group)
- Thesis title: *Event-Based Machine Vision*

MSc, Interdisciplinary Sciences **Sep 2010 - Oct 2010**

- Major Neuroscience And Physics
- Master thesis: *IFAT4 - A Dynamic Threshold Integrate-And-Fire Neuron Array Transceiver*

BSc, Interdisciplinary Sciences **Sep 2008 - Sep 2010**

- Biochemical-physical direction
- Neuroscience specialization
- Bachelor thesis: *Driving With Spikes*

PROFESSIONAL EXPERIENCE

Insightness GmbH, Zurich

Co-Founder & CEO **Since Jun 2014**

- Business development
- Sales & customer contact
- Raising money

GetYourGuide, Zurich

Backend programmer **Oct 2010 - Jan 2011**

- Development and testing of database routines
- Preparation of raw data using Smarty

Inilabs, Zurich

Software developer **Jul 2010 - Sep 2010**

- Development and testing of camera software for an interactive art installation in the train station Aarau
- Implementation of a people tracking algorithm
- Setup of an UDP based interface to a visualization software

Swiss Students Union, Bern

Member of the executive committee **Nov 2008 - Sep 2009**

- Organization and head of meetings of committee and commissions
- Responsibility for personnel administration
- Responsibility for financial management
- Responsibility for IT

Academic Experience

ETH Zurich, Institute of Neuroinformatics

Teaching Assistant
Neuromorphic Engineering I&II, 2011

- Preparation and supervision of weekly experimental classes

ETH Zurich, Chemistry Departement

Teaching Assistant
Thermodynamics for biology students, Fall 2007

- Preparation of weekly classes
- Correction of weekly exercises

Exam Preparation
Physical chemistry for chemistry students, Summer 2008
Chemistry for environmental science students, Summer 2008

- Summarization of course content and explanation of it
- Discussion of old exams

Research Experience

Sensors Group, Institute of Neuroinformatics, ETH Zurich

PhD Thesis **Since Feb 2011**

- Goal: Design of an event-based vision sensor sensitive to temporal and spatial contrast. Development of algorithms exploiting the advantages of event-based sensors
- Circuit design, simulation and layout of the sensor
- Firmware & software development for the sensor data acquisition
- Algorithm development and implementation for efficient even-base processing
- Creation and maintenance of the group homepage

Computational Sensory-Motor Systems Lab, Johns Hopkins University
Baltimore, USA

Master Thesis **Winter/spring 2010**

- Goal: Creation of an event-based integrate-and-fire neuron array transceiver(IFAT).
- Circuit design of the neuron array
- Chip layout for the neuron array chips
- Logic design for the communication with and in between the chips
- Printed circuit board design and soldering
- FPGA and soft-CPU programming

AI Lab, University of Zurich

Semester Project **Summer 2009**

- Goal: Implementation of bio-inspired reflex behaviors in the physical model of an anthropomimetic robot (CRONOS).
- Development of a framework for neural networks and evolutionary training
- Finding the setup for a evolution of feasible reflex networks

Institute Of Neuroinformatics, ETH Zurich

Bachelor Thesis **Spring 2008**

- Goal: Realization of a RC car following a lane using the information from a dynamic vision sensor (DVS).
- Development of an event-based algorithm architecture
- Implementation of neuro-inspired processing functions to extract the lane parameters

Technical Skills

Applications
Good: MATLAB, Cadence, Microsoft Office, LaTeX, Adobe Photoshop, Adobe Illustrator
Basics: Mathematica, R, Tanner CAD Tools (& SPICE), AltiumDesigner, Xilix EDK

Programming
Good: Java, VHDL
Basics: C++, C, SKILL, PHP, SQL, HTML, UNIX script

Extracurricular Courses

Venture Challenge
Successful participation including business plan development as group leader.

Extracurricular Activities

Rugby
Player since 2006, Founder of Rugby Union Zurich (RUZ) 2009, Captain 2009-2012, President since 2012

Web design
Web design in collaboration with my brother (graphics student at Zurich University of Arts).

Boy scouts
Boy scout 1991-2006, Group Leader 2000-2006, J&S leader I&II, Organizing Camps and Events as leader

Languages

Native: German

Fluent: English

Good: French

Basics: Dutch

References

Prof. Tobi Delbruck
Sensors Group
Institute of Neuroinformatics
ETH Zurich

Prof. Ralph Etienne-Cummings
Computational Sensory-Motor Systems Lab
Johns Hopkins University
Baltimore, USA

Dr. Hugo Gravato Marques
Artificial Intelligence Lab
University of Zurich
Zurich, Switzerland

A.3 Documentation

A.3.1 State Machine Diagrams

In the following the DAVIS-specific state machine diagrams used for the CPLD logic can be found. The other state machine diagrams of the used CPLD logic can be found in (Berner 2011).

A.3.1.1 Rolling Shutter Column State Machine

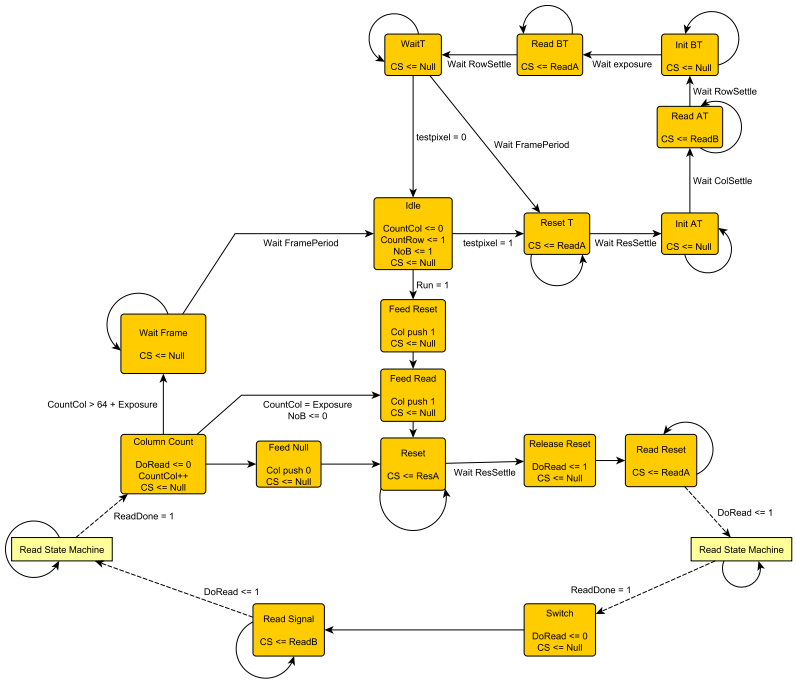


Figure A.1: Rolling Shutter Column State Machine: First the reset column pattern 110 is shifted into the shift register, then the reset column is held in reset for ResSettle clock cycles, the reset column is read and then the signal column. In the end the counter is increased and the shift register clocked. If the column counter reaches the exposure, the signal read pattern is shifted into the shift register.

A.3.1.2 Global Shutter Column State Machine

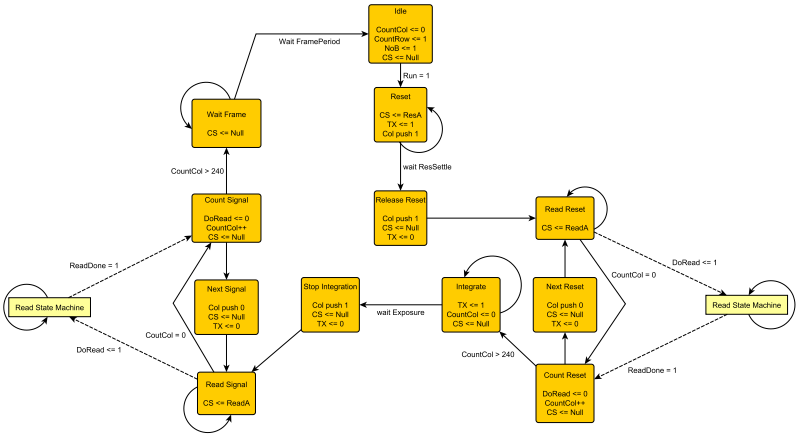


Figure A.2: Global Shutter Column State Machine: First the whole array is reset and the reset voltage is sampled, then the signal column pattern 010 is shifted into the shift register and column by column the full reset frame is acquired. Turing the exposure the TX gate is opened and the photo current is integrated. A new signal pattern is shifted into the shift register and the full signal frame is acquired.

A.3.1.3 Row State Machine

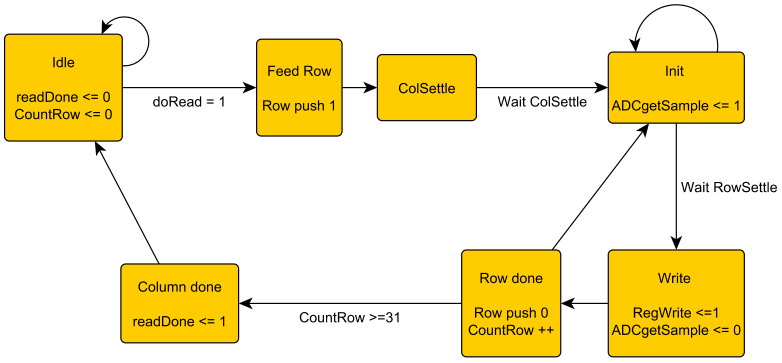


Figure A.3: Row State Machine: After a column is selected, a 1 is pushed into the register and the analog voltages in the column can settle during the ColSettle time. For each row during the RowSettle time ADC a samples are acquired and the last one is then written to the event output buffer. As soon as a full column is read, this is signaled to the column state machine

A.3.2 Pixel Variants

The positive feedback, state-holding pixel developed by Berner is shown in Fig.A.4. The pixel variant

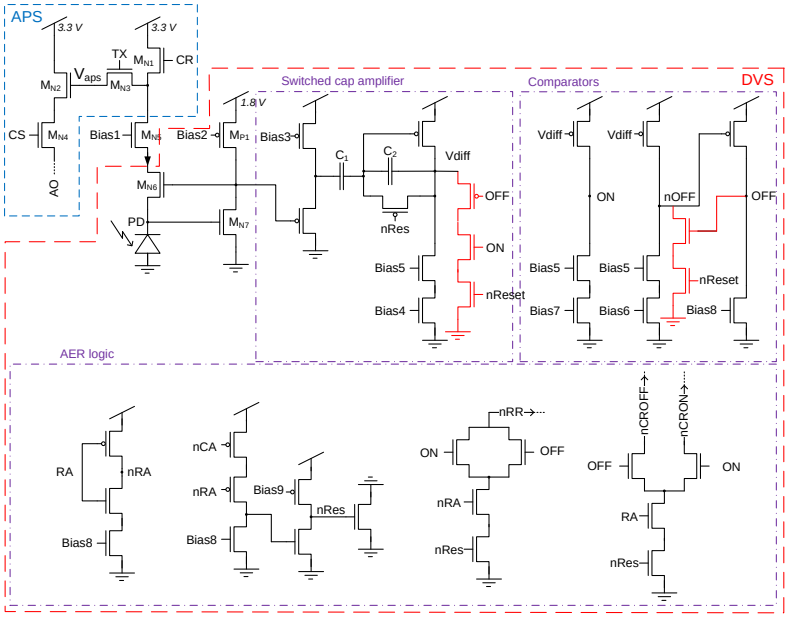


Figure A.4: Schematic of the positive feedback, state-holding pixel. The added transistors for the pull-down circuits are shown in red.

with more photoreceptor gain using a diode-connected nMOS is shown in Fig.A.5.

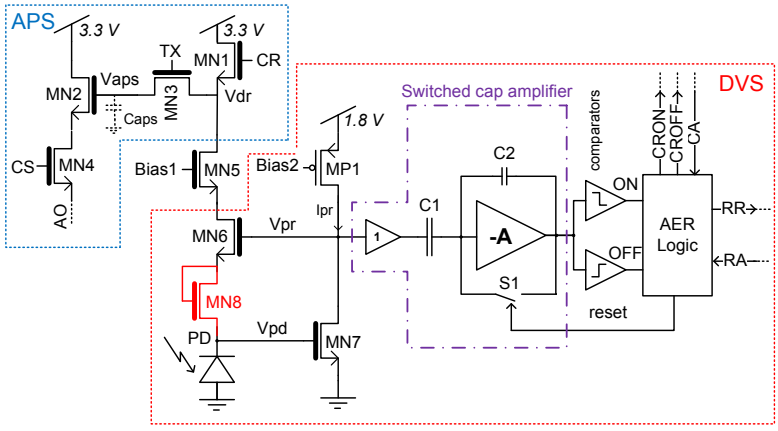


Figure A.5: Schematic of the pixel with increased photoreceptor gain. The added transistor is highlighted in red. The 3.3V transistors have thicker gates.

The layout variations of the "less GS" and "more diff gain" pixel are shown in Fig.A.6.

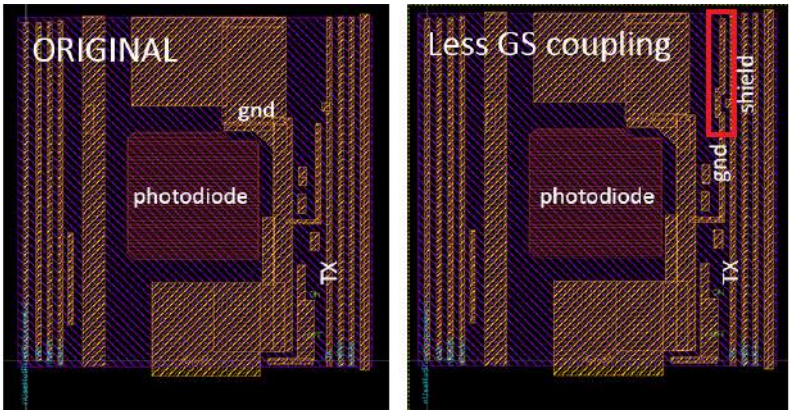


Figure A.6: M4 layout of the "less GS" pixel also used in the "more pr gain" pixel. The ground (gnd) shield is highlighted with a red box.

Bibliography

Adaptive filtering of DVS pulsed laser line response for terrain surface reconstruction (2013). url: https://www.youtube.com/watch?v=20OGD5Wwe9Q&feature=youtube_gdata_player (visited on 12/14/2014).

Alahi, A., R. Ortiz, and P. Vandergheynst (2012). "FREAK: Fast Retina Keypoint". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 510–517.

Andreou, A., K. Strohbehn, and R. E. Jenkins (1991). "Silicon retina for motion computation". In: *, IEEE International Sympoisum on Circuits and Systems, 1991*, 1373–1376 vol.3.

Arandjelovic, R. and A. Zisserman (2012). "Three things everyone should know to improve object retrieval". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2911–2918.

Arreguit, X., F. Van Schaik, F. Bauduin, M. Bidiville, and E. Raeber (1996). "A CMOS motion detector system for pointing devices". In: *Solid-State Circuits Conference, 1996. Digest of Technical Papers. 42nd ISSCC., 1996 IEEE International*, pp. 98–99.

Awatramani, G. B. and M. M. Slaughter (2000). "Origin of Transient and Sustained Responses in Ganglion Cells of the Retina". en. In: *The Journal of Neuroscience* 20.18, pp. 7087–7095. url: http://www.jneurosci.org/content/20/18/7087 (visited on 08/18/2014).

Azadmehr, M., J. P. Abrahamsen, and P. Hafliger (2005). "A foveated AER imager chip [address event representation]". In: *IEEE International Symposium on Circuits and Systems, 2005. ISCAS 2005*, 2751–2754 Vol. 3.

Bair, W. and C. Koch (1991). "Real-time motion detection using an analog VLSI zero-crossing chip". In: vol. 1473, pp. 59–65. url: http://dx.doi.org/10.1117/12.45541 (visited on 05/20/2014).

Barbaro, M., P.-Y. Burgi, A. Mortara, P. Nussbaum, and F. Heitger (2002). "A 100 x 100 pixel silicon retina for gradient extraction with steering filter capabilities and temporal output coding". In: *IEEE Journal of Solid-State Circuits* 37.2, pp. 160–172. url: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=982422 (visited on 09/30/2014).

Bartolozzi, C., F. Rea, C. Clercq, M. Hofstatter, D. Fasnacht, G. Indiveri, and G. Metta (2011a). "Embedded neuromorphic vision for humanoid robots". In: *2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 129–135.

Bartolozzi, C. and G. Indiveri (2009). "Selective Attention in Multi-Chip Address-Event Systems". en. In: *Sensors* 9.7, pp. 5076–5098. url: http://www.mdpi.com/1424-8220/9/7/5076 (visited on 04/29/2014).

Bartolozzi, C., C. Clercq, N. Mandloi, F. Rea, G. Indiveri, D. Fasnacht, G. Metta, M. Hofstätter, and R. Benosman (2011b). "eMorph: Towards Neuromorphic Robotic Vision". In: *Procedia Computer Science*. Proceedings of the 2nd European Future Technologies Conference and Exhibition 2011 (FET 11)

7, pp. 163–165. url: http://www.sciencedirect.com/science/article/pii/S1877050911005874 (visited on 04/29/2014).

Batchelor, B. G. (1999). "Coming to terms with machine vision and computer vision: they're not the same!" In: *Advanced Imaging*, pp. 22–26. url: http://www.highbeam.com/doc/1G1-53973098.html (visited on 04/02/2014).

Bauer, D., A. N. Belbachir, N. Donath, G. Gritsch, B. Kohn, M. Litzenberger, C. Posch, P. Schön, and S. Schraml (2007). "Embedded Vehicle Speed Estimation System Using an Asynchronous Temporal Contrast Vision Sensor". en. In: *EURASIP Journal on Embedded Systems* 2007.1, pp. 34–34. url: http://dx.doi.org/10.1155/2007/82174 (visited on 04/29/2014).

Bay, H., A. Ess, T. Tuytelaars, and L. Van Gool (2008). "Speeded-Up Robust Features (SURF)". In: *Computer Vision and Image Understanding*. Similarity Matching in Computer Vision and Multimedia 110.3, pp. 346–359. url: http://www.sciencedirect.com/science/article/pii/S1077314207001555 (visited on 11/21/2014).

Belbachir, A., K. Reisinger, G. Gritsch, P. Schon, and H. Garn (2007). "Automated Vehicle Velocity Estimation Using a Dual-Line Asynchronous Sensor". In: *IEEE Intelligent Transportation Systems Conference, 2007. ITSC 2007*, pp. 552–557.

Belbachir, A., M. Hofstatter, and P. Schon (2010a). "An event-driven system for high-speed vision". In: *2010 The 7th International Conference on Informatics and Systems (INFOS)*, pp. 1–5.

Belbachir, A., S. Schraml, and N. Brandle (2010b). "Real-time classification of pedestrians and cyclists for intelligent counting of non-motorized traffic". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 45–50.

Belbachir, A., A. Nowakowska, S. Schraml, G. Wiesmann, and R. Sablatnig (2011a). "Event-driven feature analysis in a 4D spatiotemporal representation for ambient assisted living". In: *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 1570–1577.

Belbachir, A., S. Schraml, and A. Nowakowska (2011b). "Event-driven stereo vision for fall detection". In: *2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 78–83.

Belbachir, A., M. Hofstatter, M. Litzenberger, and P. Scho?n (2011c). "High-Speed Embedded-Object Analysis Using a Dual-Line Timed-Address-Event Temporal-Contrast Vision Sensor". In: *IEEE Transactions on Industrial Electronics* 58.3, pp. 770–783.

Belbachir, A., M. Litzenberger, S. Schraml, M. Hofstatter, D. Bauer, P. Schon, M. Humenberger, C. Sulzbachner, T. Lunden, and M. Merne (2012a). "CARE: A dynamic stereo vision sensor system for fall detection". In: *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 731–734.

Belbachir, A., M. Mayerhofer, D. Matolin, and J. Colineau (2012b). "Real-time 360° panoramic views using BiCa360, the fast rotating dynamic vision sensor to up to 10 rotations per Sec". In: *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 727–730.

Belbachir, A. N., R. Pflugfelder, and R. Gmeiner (2010c). "A Neuromorphic Smart Camera for Real-time 360° Distortion-free Panoramas". In: *Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras*. ICDSC '10. New York, NY, USA: ACM, pp. 221–226. url: `http://doi.acm.org/10.1145/1865987.1866022` (visited on 04/29/2014).

Belbachir, A. N., S. Schraml, M. Mayerhofer, and M. Hofstatter (2014). "A Novel HDR Depth Camera for Real-Time 3D 360° Panoramic Vision". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 425–432.

Bengio, Y. (2009). *Learning Deep Architectures for AI*. en. Now Publishers Inc.

Benosman, R., S.-H. Ieng, P. Rogister, and C. Posch (2011). "Asynchronous Event-Based Hebbian Epipolar Geometry". In: *IEEE Transactions on Neural Networks* 22.11, pp. 1723–1734.

Benosman, R., C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi (2014). "Event-Based Visual Flow". In: *IEEE Transactions on Neural Networks and Learning Systems* 25.2, pp. 407–417.

Benosman, R., S.-H. Ieng, C. Clercq, C. Bartolozzi, and M. Srinivasan (2012). "Asynchronous frameless event-based optical flow". In: *Neural Networks* 27, pp. 32–37. url: `http://www.sciencedirect.com/science/article/pii/S0893608011002930` (visited on 04/24/2014).

Benson, R. G. and T. Delbrück (1991). "Direction selective silicon retina that uses null inhibition". In: *NIPS*, pp. 756–763. url: `http://www182.ini.unizh.ch/~tobi/anaprose/amon/amon.pdf` (visited on 04/24/2014).

Berlinski, D. (2001). *The Advent of the Algorithm: The 300-Year Journey from an Idea to the Computer*. English. 1 edition. New York: Mariner Books.

Berner, R., P. Lichtsteiner, and T. Delbruck (2008). "Self-timed vertacolor dichromatic vision sensor for low power pattern detection". In: *IEEE International Symposium on Circuits and Systems, 2008. ISCAS 2008*, pp. 1032–1035.

Berner, R. and T. Delbruck (2010). "Event-based color change pixel in standard CMOS". In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 349–352.

Berner, R. and T. Delbruck (2011). "Event-Based Pixel Sensitive to Changes of Color and Brightness". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 58.7, pp. 1581–1590.

Berner, R., P. Lichtsteiner, T. Delbruck, J. Kim, K. Lee, J. Lee, K. Park, T. Kim, and H. Ryu (2014). "Dynamic vision sensor for low power applications". In: *The 18th IEEE International Symposium on Consumer Electronics (ISCE 2014)*, pp. 1–2.

Berner, R. (2011). "Building Blocks for Event-Based Sensors". en. PhD Thesis. Zurich, Switzerland: ETH Zurich. url: `www.ini.uzh.ch/~tobi/wiki/lib/exe/fetch.php?media=bernerphdthesis2011.pdf`.

Berner, R. and T. Delbruck. "Photoarray combining sampled brightness sensing with asynchronous detection of time-dependent image data". Pat.

Berner, R., C. Brandli, M. Yang, S.-C. Liu, and T. Delbruck (2013a). "A 240 x 180 120dB 10mW 12us-Latency Sparse Output Vision Sensor for Mobile Applications". English. In: *2013 International Image Sensors Workshop (IISW)*. Snowbird, UT, USA. url: `http://www.imagesensors.org/Past%20Workshops/2013%20Workshop/2013%20Papers/03-1_005-delbruck_paper.pdf` (visited on 09/24/2013).

Berner, R., C. Brandli, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck (2013b). "A 240x180 10mW 12us Latency Sparse-Output Vision Sensor for Mobile Applications". In: *2013 Symposium on VLSI Circuits (VLSIC)*. Kyoto Japan, pp. C186–C187.

Bichler, O., D. Querlioz, S. Thorpe, J.-P. Bourgoin, and C. Gamrat (2011). "Unsupervised features extraction from asynchronous silicon retina through Spike-Timing-Dependent Plasticity". In: *The 2011 International Joint Conference on Neural Networks (IJCNN)*, pp. 859–866.

Bichler, O., D. Querlioz, S. J. Thorpe, J.-P. Bourgoin, and C. Gamrat (2012a). "Extraction of temporally correlated features from dynamic vision sensors with spike-timing-dependent plasticity". In: *Neural Networks*. Selected Papers from IJCNN 2011 32, pp. 339–348.

Bichler, O., M. Suri, D. Querlioz, D. Vuillaume, B. DeSalvo, and C. Gamrat (2012b). "Visual Pattern Extraction Using Energy-Efficient "2-PCM Synapse" Neuromorphic Architecture". In: *IEEE Transactions on Electron Devices* 59.8, pp. 2206–2214.

Boahen, K. A. (1996a). "Retinomorphic Vision Systems i: Pixel Design". In: *1996 IEEE International Symposium on Circuits and Systems, 1996. ISCAS '96., Connecting the World*. Vol. Supplement, pp. 9–13.

Boahen, K. A. (1996b). "Retinomorphic Vision Systems ii: Communication Channel Design". In: *, 1996 IEEE International Symposium on Circuits and Systems, 1996. ISCAS '96., Connecting the World*. Vol. Supplement, pp. 14–17.

Boahen, K. A. (1996c). "Retinomorphic vision systems". In: *Proceedings of Fifth International Conference on Microelectronics for Neural Networks, 1996*, pp. 2–14.

Boahen, K. A. (1997). "Retinomorphic vision systems : reverse engineering the vertebrate retina". phd. California Institute of Technology. url: `http://resolver.caltech.edu/CaltechETD:etd-01092008-085128` (visited on 09/07/2014).

Boahen, K. A. (2007). *A computer that works like the brain*. en. url: `http://www.ted.com/talks/kwabena_boahen_on_a_computer_that_works_like_the_brain?language=en` (visited on 09/30/2014).

Boahen, K. and A. G. Andreou (1992). "A Contrast Sensitive Silicon Retina with Reciprocal Synapses". In: *Advances in Neural Information Processing Systems 4*. Ed. by J. E. Moody, S. J. Hanson, and R. P. Lippmann. Morgan-Kaufmann, pp. 764–772. url: `http://papers.nips.cc/paper/466-a-contrast-sensitive-silicon-retina-with-reciprocal-synapses.pdf` (visited on 04/24/2014).

Boerlin, M., T. Delbruck, and K. Eng (2009). "Getting to know your neighbors: unsupervised learning of topography from real-world, event-based input". eng. In: *Neural Computation* 21.1, pp. 216–238.

Bolopion, A., Z. Ni, J. Agnus, R. Benosman, and S. Regnier (2012). "Stable haptic feedback based on a dynamic vision sensor for microrobotics". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3203–3208.

Bömmels, R., M. Machacek, A. Landolt, and T. Rösgen (2004). "Quantitative Flow Visualization for Large Scale Wind Tunnels". en. In: *The Aerodynamics of Heavy Vehicles: Trucks, Buses, and Trains*. Ed. by R. McCallen, F. Browand, and D. J. Ross. Lecture Notes in Applied and Computational Mechanics 19. Springer Berlin Heidelberg, pp. 157–167. url: `http://link.springer.com/chapter/10.1007/978-3-540-44419-0_17` (visited on 12/14/2014).

Borer, D. J. (2014). "4D Flow Visualization with Dynamic Vision Sensors". English. PhD Thesis. Zurich, Switzerland: ETH Zurich.

Bradski, G. (2000). "OpenCV". en. In: *Dr. Dobb's Journal of Software Tools*. url: `http://code.opencv.org/projects/opencv/wiki`.

Bradski, G. and A. Kaehler (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. English. 1st edition. Sebastopol, CA: O'Reilly Media.

Brandli, C., R. Berner, M. Yang, S.-C. Liu, and T. Delbruck (2014a). "A 240 x 180 130 dB 3 us Latency Global Shutter Spatiotemporal Vision Sensor". In: *IEEE Journal of Solid-State Circuits* Early Access Online.

Brandli, C. (2008). "Driving With Spikes: Lane Detection in a Semi-Neuroinspired Manner". en. Bachelor Thesis. Zurich, Switzerland: ETH Zurich. url: `www.ini.uzh.ch/~tobi/wiki/lib/exe/fetch.php?media=braendlidrivingwithspikes2008.pdf`.

Brandli, C. (2010). "IFAT4 - A Dynamic Threshold Integrate and Fire Neuron Array Transciever". en. Master Thesis. Baltimore, MD, USA: Johns Hopkins Unviersity / ETH Zurich.

Brandli, C., T. Delbruck, M. Höpflinger, M. Hutter, and T. Mantel. "Method for reconstructing a surface using spatially structured light and a dynamic vision sensor". Pat.

Brandli, C., M. Osswald, and T. Delbruck. "Method for tracking keypoints in a scene". Pat.

Brandli, C., T. Mantel, M. Hutter, M. Höpflinger, R. Berner, R. Siegwart, and T. Delbruck (2014b). "Adaptive Pulsed Laser Line Extraction for Terrain Reconstruction using a Dynamic Vision Sensor". In: *Frontiers in Neuromorphic Engineering* 7, p. 275. url: `http://www.frontiersin.org/Journal/10.3389/fnins.2013.00275/abstract` (visited on 01/04/2014).

Brandli, C., L. Muller, and T. Delbruck (2014c). "Real-Time, High-Speed Video Decompression Using a Frame- and Event-Based DAVIS Sensor". In: *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 686–689.

Brink, S., J. Hasler, and R. Wunderlich (2013). "A large-scale FPAA enabling adaptive floating-gate circuits". In: *2013 IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 285–288.

Calonder, M., V. Lepetit, C. Strecha, and P. Fua (2010). "BRIEF: Binary Robust Independent Elementary Features". en. In: *Computer Vision – ECCV 2010*. Ed. by K. Daniilidis, P. Maragos, and N. Paragios. Lecture Notes

in Computer Science 6314. Springer Berlin Heidelberg, pp. 778–792. url: http://link.springer.com/chapter/10.1007/978-3-642-15561-1_56 (visited on 11/21/2014).

Camunas-Mesa, L., A. Acosta-Jimenez, T. Serrano-Gotarredona, and B. Linares-Barranco (2008). "Fully digital AER convolution chip for vision processing". In: *IEEE International Symposium on Circuits and Systems, 2008. ISCAS 2008*, pp. 652–655.

Camuñas-Mesa, L., A. Linares-Barranco, A. J. Acosta, T. Serrano-Gotarredona, and B. Linares-Barranco (2009). "Improved AER convolution chip for vision processing with higher resolution and new functionalities". eng. In: Comunicación presentada al: "DCIS'09" celebrado en Zaragoza y organizado por la Universidad de Zaragoza (Unizar) del 18 al 20 de Noviembre del 2009. url: http://digital.csic.es/handle/10261/87071 (visited on 04/29/2014).

Camunas-Mesa, L., A. Acosta-Jimenez, C. Zamarrefio-Ramos, T. Serrano-Gotarredona, and B. Linares-Barranco (2011). "A 32 x 32 Pixel Convolution Processor Chip for Address Event Vision Sensors With 155 ns Event Latency and 20 Meps Throughput". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 58.4, pp. 777–790.

Camunas-Mesa, L., C. Zamarreno-Ramos, A. Linares-Barranco, A. Acosta-Jimenez, T. Serrano-Gotarredona, and B. Linares-Barranco (2012). "An Event-Driven Multi-Kernel Convolution Processor Module for Event-Driven Vision Sensors". In: *IEEE Journal of Solid-State Circuits* 47.2, pp. 504–517.

Camunas-Mesa, L. A., T. Serrano-Gotarredona, S. H. Ieng, R. B. Benosman, and B. Linares-Barranco (2014). "On the use of orientation filters for 3D reconstruction in event-driven stereo vision". In: *Frontiers in Neuroscience* 8. url: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3978326/ (visited on 04/29/2014).

Cardinale, J. (2006). "Tracking objects and wing beat analysis methods of a fruit fly with the event-based silicon retina". en. Semester Thesis. Zurich, Switzerland: ETH Zurich. url: http://www.ini.uzh.ch/~tobi/studentProjectReports/cardinaleWingTracker2006.pdf.

Carey, S. J., D. R. W. Barr, and P. Dudek (2013). "Low power high-performance smart camera system based on SCAMP vision sensor". In: *Journal of Systems Architecture*. Smart Camera Architecture 59.10, Part A, pp. 889–899. url: http://www.sciencedirect.com/science/article/pii/S1383762113000490 (visited on 04/29/2014).

Carneiro, J., S.-H. Ieng, C. Posch, and R. Benosman (2013). "Event-based 3D reconstruction from neuromorphic retinas". In: *Neural Networks*. Neuromorphic Engineering: From Neural Systems to Brain-Like Engineered Systems 45, pp. 27–38. url: http://www.sciencedirect.com/science/article/pii/S0893608013000725 (visited on 04/29/2014).

Casale-Rossi, M. (2014). "The heritage of Mead amp; Conway: What has remained the same, what has changed, what was missed, and what lies ahead [point of view]". In: *Proceedings of the IEEE* 102.2, pp. 114–119.

Cassidy, A. and A. Andreou (2012). "Beyond Amdahl's Law: An Objective Function That Links Multiprocessor Performance Gains to Delay and Energy". In: *IEEE Transactions on Computers* 61.8, pp. 1110–1126.

Censi, A. (2015). "Efficient Neuromorphic Optomotor Heading Regulation". en. In: *The 2015 American Control Conference*. Chicago, USA: AACC. url: `http://censi.mit.edu/pub/research/2015-neucontrol-sub.pdf`.

Censi, A., J. Strubel, C. Brandli, T. Delbruck, and D. Scaramuzza (2013). "Low-latency localization by active LED markers tracking using a dynamic vision sensor". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 891–898.

Censi, A. and D. Scaramuzza (2014). "Low-Latency Event-Based Visual Odometry". In: *IEEE International Conference on Robotics and Automation (ICRA 2014)*. Hong Kong, China.

Chan, V., C. Jin, and A van Schaik (2007). "An Address-Event Vision Sensor for Multiple Transient Object Detection". In: *IEEE Transactions on Biomedical Circuits and Systems* 1.4, pp. 278–288.

Chen, S., P. Akselrod, and E. Culurciello (2009). "A biologically inspired system for human posture recognition". In: *IEEE Biomedical Circuits and Systems Conference, 2009. BioCAS 2009*, pp. 113–116.

Chen, S., P. Akselrod, B. Zhao, J. Perez-Carrasco, B. Linares-Barranco, and E. Culurciello (2012). "Efficient Feedforward Categorization of Objects and Human Postures with Address-Event Image Sensors". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.2, pp. 302–314.

Chiang, C.-T. and C.-Y. Wu (2004). "Implantable neuromorphic vision chips". en. In: *Electronics Letters* 40.6, p. 361. url: `http://digital-library.theiet.org/content/journals/10.1049/el_20040269` (visited on 04/29/2014).

Chicca, E., P. Lichtsteiner, T. Delbruck, G. Indiveri, and R. Douglas (2006). "Modeling orientation selectivity using a neuromorphic multi-chip system". In: *2006 IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006. Proceedings*, 4 pp.–.

Chicca, E., A. Whatley, P. Lichtsteiner, V. Dante, T. Delbruck, P. Del Giudice, R. Douglas, and G. Indiveri (2007). "A Multichip Pulse-Based Neuromorphic Infrastructure and Its Application to a Model of Orientation Selectivity". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 54.5, pp. 981–993.

Clady, X., C. Clercq, S.-H. Ieng, F. Houseini, M. Randazzo, L. Natale, C. Bartolozzi, and R. B. Benosman (2014). "Asynchronous visual event-based time-to-contact". In: *Neuromorphic Engineering* 8, p. 9. url: `http://journal.frontiersin.org/Journal/10.3389/fnins.2014.00009/abstract` (visited on 04/29/2014).

Conradt, J., R. Berner, M. Cook, and T. Delbruck (2009a). "An embedded AER dynamic vision sensor for low-latency pole balancing". In: *2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 780–785.

Conradt, J., M. Cook, R. Berner, P. Lichtsteiner, R. Douglas, and T. Delbruck (2009b). "Live demonstration: A pencil balancing robot using a pair of

AER dynamic vision sensors". In: *IEEE International Symposium on Circuits and Systems, 2009. ISCAS 2009*, pp. 785–785.

Conradt, J., M. Cook, R. Berner, P. Lichtsteiner, R. Douglas, and T. Delbruck (2009c). "A pencil balancing robot using a pair of AER dynamic vision sensors". In: *IEEE International Symposium on Circuits and Systems (ISCAS) 2009*. Taipei: IEEE, pp. 781–784. url: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5117867` (visited on 08/13/2013).

Cook, M., L. Gugelmann, F. Jug, C. Krautz, and A. Steger (2011). "Interacting Maps for Fast Visual Interpretation". In: *The 2011 International Joint Conference on Neural Networks (IJCNN)*, pp. 770–776.

Costas-Santos, J., T. Serrano-Gotarredona, R. Serrano-Gotarredona, and B. Linares-Barranco (2007). "A Spatial Contrast Retina With On-Chip Calibration for Neuromorphic Spike-Based AER Vision Systems". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 54.7, pp. 1444–1458.

Culurciello, E., R. Etienne-Cummings, and K. Boahen (2001a). "Arbitrated address event representation digital image sensor". In: *Solid-State Circuits Conference, 2001. Digest of Technical Papers. ISSCC. 2001 IEEE International*, pp. 92–93.

Culurciello, E., R. Etienne-Cummings, and K. Boahen (2001b). "High dynamic range, arbitrated address event representation digital imager". In: *The 2001 IEEE International Symposium on Circuits and Systems, 2001. ISCAS 2001*. Vol. 3, 505–508 vol. 2.

Culurciello, E., R. Etienne-Cummings, and K. Boahen (2003). "A biomorphic digital image sensor". In: *IEEE Journal of Solid-State Circuits* 38.2, pp. 281–294.

Culurciello, E. and R. Etienne-Cummings (2004). "Second generation of high dynamic range, arbitrated digital imager". In: *Proceedings of the 2004 International Symposium on Circuits and Systems, 2004. ISCAS '04*. Vol. 4, IV–828–31 Vol.4.

Dan, Y., J. J. Atick, and R. C. Reid (1996). "Efficient Coding of Natural Scenes in the Lateral Geniculate Nucleus: Experimental Test of a Computational Theory". en. In: *The Journal of Neuroscience* 16.10, pp. 3351–3362. url: `http://www.jneurosci.org/content/16/10/3351` (visited on 08/19/2014).

Dankers, A. A. (2014). "Analyzing road surfaces". Pat. US8825306 B2. U.S. Classification 701/48, 701/1, 701/82; International Classification B62D6/00, G06F15/00, G05D1/00, B60T7/12; Cooperative Classification B60T2210/12, B60T8/172. url: `http://www.google.com/patents/US8825306` (visited on 10/18/2014).

Dasgupta, S. (2014). *It Began with Babbage: The Genesis of Computer Science*. en. Oxford University Press.

Davis, P. M. (1958). *Computability and Unsolvability*. en. New York: Courier Corporation.

Delbruck, T. (1993a). "Silicon retina with correlation-based, velocity-tuned pixels". In: *IEEE Transactions on Neural Networks* 4.3, pp. 529–541.

Delbrück, T. and C. A. Mead (1989). "Advances in Neural Information Processing Systems 1". In: ed. by D. S. Touretzky. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 720–727. url: http://dl.acm.org/citation.cfm?id=89851.89946 (visited on 09/07/2014).

Delbruck, T. and C. Mead (1994). "Adaptive photoreceptor with wide dynamic range". In: *, 1994 IEEE International Symposium on Circuits and Systems, 1994. ISCAS '94*. Vol. 4, 339–342 vol.4.

Delbruck, T. and P. Lichtsteiner (2006). "Fully programmable bias current generator with 24 bit resolution per bias". In: *2006 IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006. Proceedings*, 4 pp.–2852.

Delbruck, T., R. Berner, P. Lichtsteiner, and C. Dualibe (2010a). "32-bit Configurable bias current generator with sub-off-current capability". In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1647–1650.

Delbruck, T. and R. Berner (2010b). "Temporal contrast AER pixel with 0.3%-contrast event threshold". In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2442–2445.

Delbruck, T., V. Villanueva, and L. Longinotti (2014). "Integration of dynamic vision sensor with inertial measurement unit for electronically stabilized event-based vision". In: *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2636–2639.

Delbruck, T. (2008). "Frame-free dynamic digital vision". In: *Proceedings of Intl. Symp. on Secure-Life Electronics*. Tokyo, Japan. url: www.ini.uzh.ch/admin/extras/doc_get.php?id=42508.

Delbruck, T. (2012). "Fun with Asynchronous Vision Sensors and Processing". en. In: *Computer Vision – ECCV 2012. Workshops and Demonstrations*. Ed. by A. Fusiello, V. Murino, and R. Cucchiara. Lecture Notes in Computer Science 7583. Springer Berlin Heidelberg, pp. 506–515. url: http://link.springer.com/chapter/10.1007/978-3-642-33863-2_52 (visited on 04/29/2014).

Delbruck, T. and C. A. Mead (1991). "Time-derivative adaptive silicon photoreceptor array". In: ed. by T. S. J. Jayadev, pp. 92–99. url: http://spie.org/Publications/Proceedings/Paper/10.1117/12.49323 (visited on 04/24/2014).

Delbrück, T. and C. A. Mead (1996). "Analog VLSI phototransduction". In: *CNS Memo* 30, p. 10. url: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.156.2987&rep=rep1&type=pdf (visited on 04/24/2014).

Delbruck, T. and P. Lichtsteiner (2007). "Fast Sensory Motor Control Based on Event-Based Hybrid Neuromorphic-Procedural System". In: IEEE, pp. 845–848. url: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4252767 (visited on 08/13/2013).

Delbruck, T., B. Linares-Barranco, E. Culurciello, and C. Posch (2010c). "Activity-Driven, Event-Based Vision Sensors". In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2426–2429.

Delbruck, T. and M. Lang (2013). "Robotic goalie with 3 ms reaction time at 4% CPU load using event-based dynamic vision sensor". In: *Frontiers in*

Neuroscience 7. url: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3836084/ (visited on 04/29/2014).

Delbruck, T. (1993b). "Investigations of analog VLSI visual transduction and motion processing". phd. California Institute of Technology. url: http://resolver.caltech.edu/CaltechETD:etd-07022004-144710 (visited on 09/08/2014).

Denk, C., F. Llobet-Blandino, F. Galluppi, L. A. Plana, S. Furber, and J. Conradt (2013). "Real-Time Interface Board for Closed-Loop Robotic Tasks on the SpiNNaker Neural Computing System". In: *Artificial Neural Networks and Machine Learning – ICANN 2013*. Ed. by V. Mladenov, P. Koprinkova-Hristova, G. Palm, A. E. P. Villa, B. Appollini, and N. Kasabov. Lecture Notes in Computer Science 8131. Springer Berlin Heidelberg, pp. 467–474. url: http://link.springer.com/chapter/10.1007/978-3-642-40728-4_59 (visited on 04/29/2014).

Domi?nguez-Morales, M., A. Jimenez-Fernandez, R. Paz-Vicente, G. Jime?nez, and A. Linares-Barranco (2012). "Live demonstration: On the distance estimation of moving targets with a Stereo-Vision AER system". In: *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 721–725.

Dominguez-Morales, M. J., E. Cerezuela-Escudero, F. Perez-Pena, A. Jimenez-Fernandez, A. Linares-Barranco, and G. Jimenez-Moreno (2013). "On the AER Stereo-Vision Processing: A Spike Approach to Epipolar Matching". In: *Neural Information Processing*. Ed. by M. Lee, A. Hirose, Z.-G. Hou, and R. M. Kil. Lecture Notes in Computer Science 8226. Springer Berlin Heidelberg, pp. 267–275. url: http://link.springer.com/chapter/10.1007/978-3-642-42054-2_34 (visited on 04/29/2014).

Drazen, D., P. Lichtsteiner, P. Häfliger, T. Delbrück, and A. Jensen (2011). "Toward real-time particle tracking using an event-based dynamic vision sensor". en. In: *Experiments in Fluids* 51.5, pp. 1465–1469. url: http://link.springer.com/article/10.1007/s00348-011-1207-y (visited on 04/23/2014).

Drubach, D. (1999). *The Brain Explained*. English. 1 edition. Upper Saddle River, NJ: Prentice Hall.

Dudek, P. (2005). "Implementation of SIMD vision chip with 128 times;128 array of analogue processing elements". In: *IEEE International Symposium on Circuits and Systems, 2005. ISCAS 2005*, 5806–5809 Vol. 6.

Dudek, P. and P. Hicks (2001). "A general-purpose CMOS vision chip with a processor-per-pixel SIMD array". In: *Solid-State Circuits Conference, 2001. ESSCIRC 2001. Proceedings of the 27th European*, pp. 213–216.

Etienne-Cummings, R. and J. Van der Spiegel (1996). "Neuromorphic vision sensors". In: *Sensors and Actuators A: Physical* 56.1–2, pp. 19–29. url: http://www.sciencedirect.com/science/article/pii/0924424796012770 (visited on 04/24/2014).

Etienne-Cummings, R., J. Van der Spiegel, and P. Mueller (1997). "A focal plane visual motion measurement sensor". In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 44.1, pp. 55–66.

Etienne-Cummings, R., J. Van der Spiegel, and P. Mueller (1999). "Hardware implementation of a visual-motion pixel using oriented spatiotemporal

neural filters". In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 46.9, pp. 1121–1136.

Etienne-Cummings, R., S. Fernando, J. Van der Spiegel, and P. Mueller (1992). "Real-time 2D analog motion detector VLSI circuit". In: *, International Joint Conference on Neural Networks, 1992. IJCNN*. Vol. 4, 426–431 vol.4.

Foix, S., G. Alenya, and C. Torras (2011). "Lock-in Time-of-Flight (ToF) Cameras: A Survey". In: *IEEE Sensors Journal* 11.9, pp. 1917–1926.

Forest, J. and J. Salvi (2002). "A Review of Laser Scanning Three-Dimensional Digitisers". In: vol. 1. IEEE, pp. 73–78. url: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1041365` (visited on 08/13/2013).

Forster, C., M. Pizzoli, and D. Scaramuzza (2014). "SVO: Fast semi-direct monocular visual odometry". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15–22.

Fossum, E. R. (1993). "Active pixel sensors: are CCDs dinosaurs?" In: vol. 1900, pp. 2–14. url: `http://dx.doi.org/10.1117/12.148585` (visited on 05/19/2014).

Freeth, T., Y. Bitsakis, X. Moussas, J. H. Seiradakis, A. Tselikas, H. Mangou, M. Zafeiropoulou, R. Hadland, D. Bate, A. Ramsey, M. Allen, A. Crawley, P. Hockley, T. Malzbender, D. Gelb, W. Ambrisco, and M. G. Edmunds (2006). "Decoding the ancient Greek astronomical calculator known as the Antikythera Mechanism". en. In: *Nature* 444.7119, pp. 587–591. url: `http://www.nature.com/nature/journal/v444/n7119/abs/nature05357.html` (visited on 12/13/2014).

Fu, Z., T. Delbruck, P. Lichtsteiner, and E. Culurciello (2008). "An Address-Event Fall Detector for Assisted Living Applications". In: *IEEE Transactions on Biomedical Circuits and Systems* 2.2, pp. 88–96.

Fujii, T. and T. Fukuchi (2005). *Laser Remote Sensing*. en. CRC Press.

Fukushima, K., Y. Yamaguchi, M. Yasuda, and S. Nagata (1970). "An electronic model of the retina". In: *Proceedings of the IEEE* 58.12, pp. 1950–1951.

Galluppi, F., K. Brohan, S. Davidson, T. Serrano-Gotarredona, J.-A. P. Carrasco, B. Linares-Barranco, and S. Furber (2012). "A Real-Time, Event-Driven Neuromorphic System for Goal-Directed Attentional Selection". In: *Neural Information Processing*. Ed. by T. Huang, Z. Zeng, C. Li, and C. S. Leung. Lecture Notes in Computer Science 7664. Springer Berlin Heidelberg, pp. 226–233. url: `http://link.springer.com/chapter/10.1007/978-3-642-34481-7_28` (visited on 04/29/2014).

Garcia Franco, J., J. del Valle Padilla, and S. Ortega Cisneros (2013). "Event-based image processing using a neuromorphic vision sensor". In: *2013 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, pp. 1–6.

Gilder, G. F. (2005). *The Silicon Eye*. English. New York: W.W. Norton & Co.

Gioi, R. von, J. Jakubowicz, J.-M. Morel, and G. Randall (2010). "LSD: A Fast Line Segment Detector with a False Detection Control". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.4, pp. 722–732.

Gioi, R. Grompone von, J. Jakubowicz, J.-M. Morel, and G. Randall (2012). "LSD: a Line Segment Detector". In: *Image Processing On Line* 2, pp. 35–55.

url: http://www.ipol.im/pub/art/2012/gjmr-lsd/?utm_source=doi (visited on 11/22/2014).

Gisler, D. (2007). "Eye Tracking using Event-Based Silicon Retina". en. Semester Thesis. Zurich, Switzerland: ETH Zurich. url: http://www.ini.uzh.ch/~tobi/studentProjectReports/gislerEyeTracking2007.pdf.

Gollisch, T. and M. Meister (2010). "Eye Smarter than Scientists Believed: Neural Computations in Circuits of the Retina". In: *Neuron* 65.2, pp. 150–164. url: http://www.sciencedirect.com/science/article/pii/S0896627309009994 (visited on 08/18/2014).

Gomez-Rodriguez, F., L. Miro-Amarante, F. Diaz-del Rio, A. Linares-Barranco, and G. Jimenez (2010). "Real time multiple objects tracking based on a bio-inspired processing cascade architecture". In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1399–1402.

Gomez-Rodriguez, F., L. Miró-Amarante, M. Rivas, G. Jimenez, and F. Diaz-del Rio (2011). "Neuromorphic Real-Time Objects Tracking Using Address Event Representation and Silicon Retina". In: *Advances in Computational Intelligence*. Ed. by J. Cabestany, I. Rojas, and G. Joya. Lecture Notes in Computer Science 6691. Springer Berlin Heidelberg, pp. 133–140. url: http://link.springer.com/chapter/10.1007/978-3-642-21501-8_17 (visited on 04/29/2014).

Goodale, M. A. and A. D. Milner (1992). "Separate visual pathways for perception and action". In: *Trends in Neurosciences* 15.1, pp. 20–25.

Gorton, W. (1998). "The Genesis Of The Transistor". In: *Proceedings of the IEEE* 86.1, pp. 50–52.

Graf, R., A. Belbachir, R. King, and M. Mayerhofer (2013). "Quality control of real-time panoramic views from the smart camera 360SCAN". In: *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 650–653.

Gritsch, G., M. Litzenberger, N. Donath, and B. Kohn (2008). "Real-Time Vehicle Classification using a Smart Embedded Device with a `Silicon Retina' Optical Sensor". In: *11th International IEEE Conference on Intelligent Transportation Systems, 2008. ITSC 2008*, pp. 534–538.

Gritsch, G., N. Donath, B. Kohn, and M. Litzenberger (2009). "Night-time vehicle classification with an embedded, vision system". In: *12th International IEEE Conference on Intelligent Transportation Systems, 2009. ITSC '09*, pp. 1–6.

Gunjal, V. (2012). "Development of Feature Descriptors for Event-Based Vision Sensors". Bachelor Thesis. Zurich, Switzerland: ETH Zurich. url: www.ini.uzh.ch/~tobi/wiki/lib/exe/fetch.php?media=gunjalbachelorsthesis2012.pdf.

Guo, X., X. Qi, and J. Harris (2007). "A Time-to-First-Spike CMOS Image Sensor". In: *IEEE Sensors Journal* 7.8, pp. 1165–1175.

Haandbaek, N., K. Mathwig, R. Streichan, N. Goedecke, S. C. Bürgel, F. Heer, and A. Hierlemann (2011). "Characterization of Cell Phenotype Using Dynamic Vision Sensor and Impedance Spectroscope". en. In: *Proceeding of 15th International Conference on Miniaturized Systems for Chemistry and Life Sciences*. Seattle, Washington. url: http://www.rsc.org/images/LOC/2011/PDFs/Papers/414_1290.pdf.

Harris, C. and M. Stephens (1988). "A Combined Corner and Edge Detector". en. In: *Proceedings of the Alvey Vision Conference*. Alvey Vision Club, pp. 23.1–23.6. url: http://www.bmva.org/bmvc/1988/avc-88-023.pdf (visited on 11/21/2014).

Harrison, R. R. and C. Koch (1999). "A Robust Analog VLSI Motion Sensor Based on the Visual System of the Fly". en. In: *Autonomous Robots* 7.3, pp. 211–224. url: http://link.springer.com/article/10.1023/A%3A1008916202887 (visited on 04/24/2014).

Hasler, J. and B. Marr (2013). "Finding a roadmap to achieve large neuromorphic hardware systems". In: *Frontiers in Neuroscience* 7. url: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3767911/ (visited on 09/11/2014).

Hasler, P. and T. Lande (2001). "Overview of floating-gate devices, circuits, and systems". In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 48.1, pp. 1–3.

Hebart, M. N. and G. Hesselmann (2012). "What Visual Information Is Processed in the Human Dorsal Stream?" en. In: *The Journal of Neuroscience* 32.24, pp. 8107–8109. url: http://www.jneurosci.org/content/32/24/8107 (visited on 08/19/2014).

Hess, P. (2006). "Low-level Stereo Matching using Event-based Silicon Retina". PhD thesis. Zurich, Switzerland: ETH Zurich. url: http://www.ini.uzh.ch/~tobi/studentProjectReports/hessAERStereo2006.pdf.

Hoffmann, R., D. Weikersdorfer, and J. Conradt (2013). "Autonomous indoor exploration with an event-based visual SLAM system". In: *2013 European Conference on Mobile Robots (ECMR)*, pp. 38–43.

Hofstatter, M., P. Scho?n, and C. Posch (2009). "An integrated 20-bit 33/5M events/s AER sensor interface with 10ns time-stamping and hardware-accelerated event pre-processing". In: *IEEE Biomedical Circuits and Systems Conference, 2009. BioCAS 2009*, pp. 257–260.

Hofstetter, M. (2012). "Temporal Pattern-Based Active Marker Identification and Tracking Using a Dynamic Vision Sensor". Master Thesis. Zurich, Switzerland: ETH Zurich. url: www.ini.uzh.ch/~tobi/wiki/lib/exe/fetch.php?media=matthiashofstettersmasterthesis2012.pdf.

Horiuchi, T., W. Bair, B. Bishofberger, A. Moore, C. Koch, and J. Lazzaro (1992). "Computing motion using analog VLSI vision chips: An experimental comparison among different approaches". en. In: *International Journal of Computer Vision* 8.3, pp. 203–216. url: http://link.springer.com/article/10.1007/BF00055152 (visited on 05/20/2014).

Horn, B. K. P. and B. G. Schunck (1993). ""Determining optical flow": a retrospective". In: *Artificial Intelligence* 59.1–2, pp. 81–87. url: http://www.sciencedirect.com/science/article/pii/0004370293901739 (visited on 12/14/2014).

Horn, B. K. and B. G. Schunck (1981). "Determining Optical Flow". In: vol. 0281, pp. 319–331. url: http://dx.doi.org/10.1117/12.965761 (visited on 12/14/2014).

Hydroneuron (2012). Youtube. url: https://www.youtube.com/watch?v=yNt9sYhasnI&feature=youtube_gdata_player (visited on 11/29/2014).

Ieng, S., C. Posch, and R. Benosman (2014). "Asynchronous Neuromorphic Event-Driven Image Filtering". In: *Proceedings of the IEEE* 102.10, pp. 1485–1499.

Illingworth, J. and J. Kittler (1988). "A survey of the hough transform". In: *Computer Vision, Graphics, and Image Processing* 44.1, pp. 87–116. url: http://www.sciencedirect.com/science/article/pii/S0734189X88800331 (visited on 10/13/2014).

Indiveri, G. (1999). "Neuromorphic analog VLSI sensor for visual tracking: circuits and application examples". In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 46.11, pp. 1337–1347.

Janesick, J. R. (2007). *Photon Transfer*. 1000 20th Street, Bellingham, WA 98227-0010 USA: SPIE. url: http://ebooks.spiedigitallibrary.org/book.aspx?bookid=117 (visited on 09/26/2013).

Kahng, D. (1963). "Electric field controlled semiconductor device". Pat. US3102230 A. U.S. Classification 323/349, 330/277, 257/288, 438/586, 327/581; International Classification H01L23/29, H01L29/00; Cooperative Classification H01L29/00, H01L2924/3011, H01L23/291; European Classification H01L23/29C, H01L29/00. url: http://www.google.com/patents/US3102230 (visited on 12/13/2014).

Kandel, E. R., J. H. Schwartz, T. M. Jessell, S. A. Siegelbaum, and A. J. Hudspeth (2012). *Principles of Neural Science, Fifth Edition*. English. 5th edition. New York: McGraw-Hill Professional.

Katz, M. L., K. Nikolic, and T. Delbruck (2012). "Live demonstration: Behavioural emulation of event-based vision sensors". In: *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 736–740.

Kim, H., A. Handa, R. Benosman, S.-H. Ieng, and A. J. Davison (2014). "Simultaneous Mosaicing and Tracking with an Event Camera". en. In: *Proceedings of the British Machine Vision Conference (BMVC)*. Nottingham: BVMC Press. url: http://www.doc.ic.ac.uk/~ajd/Publications/kim_etal_bmvc2014.pdf.

Knuth, D. E. (1998). *The Art of Computer Programming, Vols. 1-3*. English. 3rd edition. Reading, Mass.: Addison-Wesley Professional.

Koch, C. (1991). "Implementing early vision algorithms in analog hardware: an overview". In: vol. 1473, pp. 2–16. url: http://dx.doi.org/10.1117/12.45546 (visited on 05/20/2014).

Koch, C. and M. Bimaul (1996). "Neuromorphic vision chips". In: *IEEE Spectrum*.

Koeth, F., H. G. Marques, and T. Delbruck (2013). "Self-organisation of motion features with a temporal asynchronous dynamic vision sensor". In: *Biologically Inspired Cognitive Architectures*. BICA 2013: Papers from the Fourth Annual Meeting of the BICA Society 6, pp. 8–11. url: http://www.sciencedirect.com/science/article/pii/S2212683X13000455 (visited on 04/29/2014).

Kogler, J., F. Eibensteiner, M. Humenberger, M. Gelautz, and J. Scharinger (2013). "Ground Truth Evaluation for Event-Based Silicon Retina Stereo Data". In: *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 649–656.

Kohn, B., A. Belbachir, T. Hahn, and H. Kaufmann (2012a). "Event-driven body motion analysis for real-time gesture recognition". In: *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 703–706.

Kohn, B., A. Belbachir, and A. Nowakowska (2012b). "Real-time gesture recognition using bio inspired 3D vision sensor". In: *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 37–42.

Kramer, J. (1996). "Compact integrated motion sensor with three-pixel interaction". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18.4, pp. 455–460.

Kramer, J. (2002a). "An integrated optical transient sensor". In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 49.9, pp. 612–628.

Kramer, J. (2002b). "An on/off transient imager with event-driven, asynchronous read-out". In: *IEEE International Symposium on Circuits and Systems, 2002. ISCAS 2002*. Vol. 2, II–165–II–168 vol.2.

Kramer, J., R. Sarpeshkar, and C. Koch (1995). "An analog VLSI velocity sensor". In: *, 1995 IEEE International Symposium on Circuits and Systems, 1995. ISCAS '95*. Vol. 1, 413–416 vol.1.

Kramer, J., R. Sarpeshkar, and C. Koch (1996). "Analog VLSI Motion Discontinuity Detectors For Image Segmentation". In: *IEEE International Symposium on Circuits and Systems (ISCAS)*.

Lagorce, X., S.-H. Ieng, and R. Benosman (2013). "Event-based features for robotic vision". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4214–4219.

Lang, M. (2007). "Learning Robotic Goalie". en. Seme. Zurich, Switzerland: ET. url: `www.ini.uzh.ch/~tobi/wiki/lib/exe/fetch.php?media=manuellangsemesterthesisss07.pdf`.

Lazar, A. and E. Pnevmatikakis (2011). "Video Time Encoding Machines". In: *IEEE Transactions on Neural Networks* 22.3, pp. 461–473.

Lazzaro, J., J. Wawrzynek, M. Mahowald, M. Sivilotti, and D. Gillespie (1993). "Silicon auditory processors as computer peripherals". In: *IEEE Trans. Neural Netw.* 4.3, pp. 523–528.

Ledak, P. J. (2011). *Who is Watson.* en. url: `ftp://ftp.software.ibm.com/la/documents/imc/la/co/swg_bogota/plenaria/watson_paul_ledack.pdf`.

Lee, J., T. Delbruck, M. Pfeiffer, P. Park, C.-W. Shin, H. Ryu, and B. Kang (2014). "Real-Time Gesture Interface Based on Event-Driven Processing from Stereo Silicon Retinas". In: *IEEE Transactions on Neural Networks and Learning Systems* PP.99, pp. 1–1.

Lee, J. H., P. Park, C.-W. Shin, H. Ryu, B. C. Kang, and T. Delbruck (2012a). "Touchless hand gesture UI with instantaneous responses". In: *2012 19th IEEE International Conference on Image Processing (ICIP)*, pp. 1957–1960.

Lee, J., T. Delbruck, P. Park, M. Pfeiffer, C.-W. Shin, H. Ryu, and B.-C. Kang (2012b). "Live demonstration: Gesture-based remote control using stereo pair of dynamic vision sensors". In: *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 741–745.

Leñero-Bardallo, J., T. Serrano-Gotarredona, and B. Linares-Barranco (2010). "A Five-Decade Dynamic-Range Ambient-Light-Independent Calibrated Signed-Spatial-Contrast AER Retina With 0.1-ms Latency and Optional Time-to-First-Spike Mode". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 57.10, pp. 2632–2643.

Lenero-Bardallo, J., D. Bryn, and P. Hafliger (2013). "Flame monitoring with an AER color vision sensor". In: *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2404–2407.

Lenero-Bardallo, J. and P. Hafliger (2014). "A dual operation mode bio-inspired pixel". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* Early Access Online.

Lenero-Bardallo, J. A., T. Serrano-Gotarredona, and B. Linares-Barranco (2010). "A signed spatial contrast event spike retina chip". In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2438–2441.

Lenero-Bardallo, J. A., T. Serrano-Gotarredona, and B. Linares-Barranco (2011a). "A 3.6 us Latency Asynchronous Frame-Free Event-Driven Dynamic-Vision-Sensor". In: *IEEE Journal of Solid-State Circuits* 46.6, pp. 1443–1455. url: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5746543` (visited on 08/13/2013).

Lenero-Bardallo, J. A., D. H. Bryn, and P. Hafliger (2011b). "Bio-inspired asynchronous pixel event tri-color vision sensor". In: *2011 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 253–256.

Lenero-Bardallo, J. A., D. H. Bryn, and P. Hafliger (2012). "Live demonstration: A bio-inspired asynchronous pixel event tri-color vision sensor". In: *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 726–726.

Leutenegger, S., M. Chli, and R. Siegwart (2011). "BRISK: Binary Robust invariant scalable keypoints". In: *2011 IEEE International Conference on Computer Vision (ICCV)*, pp. 2548–2555.

Lichtsteiner, P., T. Delbruck, and C. Posch (2006a). "A 100dB dynamic range high-speed dual-line optical transient sensor with asynchronous readout". In: *2006 IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006. Proceedings*, 4 pp.–1662.

Lichtsteiner, P., C. Posch, and T. Delbruck (2006b). "A 128 X 128 120db 30mw asynchronous vision sensor that responds to relative intensity change". In: *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, pp. 2060–2069.

Lichtsteiner, P. (2006). "A temporal contrast vision sensor". en. PhD thesis. Zurich, Switzerland: ETH Zurich. url: `www.ini.uzh.ch/~tobi/wiki/lib/exe/fetch.php?media=lichtsteinerthesis2006.pdf`.

Lichtsteiner, P., T. Delbruck, and J. Kramer (2004). "Improved ON/OFF temporaly differentiating address-event imager". en. In: *11th IEEE International Conference on Electronics, Circuits and Systems*, pp. 211–214. url: `http://www.ini.uzh.ch/~tobi/papers/lichtsteinerICECS2004.pdf`.

Lichtsteiner, P. and T. Delbruck (2005). "64x64 Event-Driven Logarithmic Temporal Derivative Silicon Retina". In: *IEEE Workshop on Charge- Coupled Devices and Advanced Image Sensors*. Nagano, Japan. url: www.ini.uzh.ch/admin/extras/doc_get.php?id=42216.

Lichtsteiner, P., C. Posch, and T. Delbruck (2008). "A 128 x 128 120 dB 15us Latency Asynchronous Temporal Contrast Vision Sensor". In: *IEEE Journal of Solid-State Circuits* 43.2, pp. 566–576. url: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4444573 (visited on 08/13/2013).

Lichtsteiner, P., CH, T. Delbruck, and CH (2010). "United States Patent: 7728269 - Photoarray for detecting time-dependent image data". Pat. 7728269. url: http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=%2Fnetahtml%2FPTO%2Fsearch-bool.html&r=1&f=G&l=50&co1=AND&d=PTXT&s1=delbruck.INNM.&OS=IN/delbruck&RS=IN/delbruck (visited on 09/16/2013).

Linares-Barranco, A., F. Gomez-Rodriguez, A. Jimenez-Fernandez, T. Delbruck, and P. Lichtensteiner (2007). "Using FPGA for visuo-motor control with a silicon retina and a humanoid robot". In: *IEEE International Symposium on Circuits and Systems, 2007. ISCAS 2007*, pp. 1192–1195.

Linares-Barranco, A., R. Paz-Vicente, F. Gomez-Rodriguez, A. Jimenez, M. Rivas, G. Jimenez, and A. Civit (2010). "On the AER convolution processors for FPGA". In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 4237–4240.

Litzenberger, M., C. Posch, D. Bauer, A. Belbachir, P. Schon, B. Kohn, and H. Garn (2006a). "Embedded Vision System for Real-Time Object Tracking using an Asynchronous Transient Vision Sensor". In: *Digital Signal Processing Workshop, 12th - Signal Processing Education Workshop, 4th*, pp. 173–178.

Litzenberger, M., A. Belbachir, N. Donath, G. Gritsch, H. Garn, B. Kohn, C. Posch, and S. Schraml (2006b). "Estimation of Vehicle Speed Based on Asynchronous Data from a Silicon Retina Optical Sensor". In: *IEEE Intelligent Transportation Systems Conference, 2006. ITSC '06*, pp. 653–658.

Litzenberger, M., A. Belbachir, P. Schon, and C. Posch (2007a). "Embedded Smart Camera for High Speed Vision". In: *First ACM/IEEE International Conference on Distributed Smart Cameras, 2007. ICDSC '07*, pp. 81–86.

Litzenberger, M., B. Kohn, G. Gritsch, N. Donath, C. Posch, N. A. Belbachir, and H. Garn (2007b). "Vehicle Counting with an Embedded Traffic Data System using an Optical Transient Sensor". In: *IEEE Intelligent Transportation Systems Conference, 2007. ITSC 2007*, pp. 36–40.

Litzenberger, S. and A. Sabo (2012). "Can Silicon Retina Sensors be used for optical motion analysis in sports?" In: *Procedia Engineering*. ENGINEERING OF SPORT CONFERENCE 2012 34, pp. 748–753. url: http://www.sciencedirect.com/science/article/pii/S1877705812017419 (visited on 04/29/2014).

Liu, S.-C., J. Kramer, G. Indiveri, T. Delbruck, and R. Douglas (2002). *Analog VLSI: Circuits and Principles*. English. Cambridge, Mass: A Bradford Book.

Liu, S.-C., T. Delbruck, G. Indiveri, A. Whatley, and R. Douglas, eds. (2014). *Event-Based Neuromorphic Systems*. John Wiley and Sons Ltd., UK.

Longinotti, L. (2014). "cAER: A framework for event-based processing on embedded systems". en. Bachelor Thesis. Zurich, Switzerland: University of Zurich.

Lopich, A. and P. Dudek (2010). "An 80 x 80 general-purpose digital vision chip in 0.18 #x03BC;m CMOS technology". In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 4257–4260.

Lopich, A. and P. Dudek (2011). "A SIMD Cellular Processor Array Vision Chip With Asynchronous Processing Capabilities". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 58.10, pp. 2420–2431.

Lowe, D. (1999). "Object recognition from local scale-invariant features". In: *The Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999*. Vol. 2, 1150–1157 vol.2.

Lowe, D. G. (2004). "Distinctive Image Features from Scale-Invariant Keypoints". en. In: *International Journal of Computer Vision* 60.2, pp. 91–110. url: `http://link.springer.com/article/10.1023/B%3AVISI.0000029664.99615.94` (visited on 11/21/2014).

Lyon, R. F. and P. M. Hubel (2002). "Eyeing the camera: Into the next century". In: *in Proc. IS&T/SID 10th Color Imaging Conf., 2002*, pp. 349–355.

MacCormick, J. (2011). *How does the Kinect work?* en. Talk. Dickinson College. url: `users.dickinson.edu/~jmac/selected-talks/kinect.pdf`.

Mahowald, M. (1994a). "Analog VLSI chip for stereocorrespondence". In: *, 1994 IEEE International Symposium on Circuits and Systems, 1994. ISCAS '94*. Vol. 6, 347–350 vol.6.

Mahowald, M. (1992). "VLSI Analogs of Neuronal Visual Processing: A Synthesis of Form and Function". en. PhD Thesis. Pasadena, CA: California Institute of Technology. url: `http://www.ini.unizh.ch/~tobi/papers/mishathesis.pdf`.

Mahowald, M. (1994b). "The Silicon Optic Nerve". en. In: *An Analog VLSI System for Stereoscopic Vision*. The Springer International Series in Engineering and Computer Science 265. Springer US, pp. 66–117. url: `http://link.springer.com/chapter/10.1007/978-1-4615-2724-4_3` (visited on 05/19/2014).

Mantel, T. A. (2012). "Dynamic vision sensor for surface reconstruction". en. Semester Thesis. Zurich: ETH Zurich. url: `http://students.asl.ethz.ch/project.php?pid=367`.

Marguin, J. (1994). *Histoire des instruments et machines à calculer: trois siècles de mécanique pensante : 1642-1942*. French. Paris: Hermann.

Marr, D. and T. Poggio (1979). "A Computational Theory of Human Stereo Vision". en. In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 204.1156, pp. 301–328. url: `http://rspb.royalsocietypublishing.org/content/204/1156/301` (visited on 10/14/2014).

Martin, A. J., A. Lines, R. Manohar, M. Nystroem, P. Penzes, R. Southworth, and U. Cummings (1997). "The Design of an Asynchronous MIPS R3000 Microprocessor". In: *Advanced Research in VLSI, Conference on*. Vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, p. 164.

Matolin, D., C. Posch, R. Wohlgenannt, and T. Maier (2008). "A 64 x 64 pixel temporal contrast microbolometer infrared sensor". In: *IEEE International Symposium on Circuits and Systems, 2008. ISCAS 2008*, pp. 1644–1647.

Matolin, D., C. Posch, and R. Wohlgenannt (2009). "True correlated double sampling and comparator design for time-based image sensors". In: *IEEE International Symposium on Circuits and Systems, 2009. ISCAS 2009*, pp. 1269–1272.

Matolin, D., R. Wohlgenannt, M. Litzenberger, and C. Posch (2010). "A load-balancing readout method for large event-based PWM imaging arrays". In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 361–364.

Mead, C. A. and M. A. Mahowald (1988). "A silicon model of early visual processing". In: *Neural Networks* 1.1, pp. 91–97. url: `http://www.sciencedirect.com/science/article/pii/089360808890024X` (visited on 04/24/2014).

Mead, C. (1985). "A sensitive electronic photoreceptor". In: *1985 Chapel Hill Conference on VLSI*. Rockville: Computer Science Press, pp. 463–471.

Mead, C. (1989). *Analog VLSI and Neural Systems*. English. 1st edition. Reading, Mass: Addison Wesley Publishing Company.

Mitchell, L. (2001). "The Man Who Stopped Time". en. In: *Standford Magazine* May/June. url: `http://alumni.stanford.edu/get/page/magazine/article/?article_id=39117`.

Monteiro, H. A. P. (2013). "Neuromorphic systems for legged robot control". en. In: url: `http://www.era.lib.ed.ac.uk/handle/1842/7736` (visited on 04/29/2014).

Moore, A. J. and C. Koch (1991). "Multiplication-based analog motion detection chip". In: vol. 1473, pp. 66–75. url: `http://dx.doi.org/10.1117/12.45542` (visited on 05/20/2014).

Mortara, A, E. Vittoz, and P. Venier (1995). "A communication scheme for analog VLSI perceptive systems". In: *IEEE Journal of Solid-State Circuits* 30.6, pp. 660–669.

Mueggler, E., B. Huber, and D. Scaramuzza (2014). "Event-based, 6-DOF pose tracking for high-speed maneuvers". In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, pp. 2761–2768.

Muller, G. and J. Conradt (2011). "A miniature low-power sensor system for real time 2D visual tracking of LED markers". In: *2011 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2429–2434.

Müller, G. R. and J. Conradt (2012). "Self-calibrating Marker Tracking in 3D with Event-Based Vision Sensors". In: *Artificial Neural Networks and Machine Learning – ICANN 2012*. Ed. by A. E. P. Villa, W. Duch, P. Érdi, F. Masulli, and G. Palm. Lecture Notes in Computer Science 7552. Springer Berlin Heidelberg, pp. 313–321. url: `http://link.springer.com/chapter/10.1007/978-3-642-33269-2_40` (visited on 04/29/2014).

Münch, T., R. Da Silveira, S. Siegert, T. Viney, G. Awatramani, and B. Roska (2009). "Approach sensitivity in the retina processed by a multifunctional neural circuit". English. In: *Nature Neuroscience* 12.10, pp. 1308–1316.

Nageswaran, J., N. Dutt, Y. Wang, and T. Delbrueck (2009). "Computing spike-based convolutions on GPUs". In: *IEEE International Symposium on Circuits and Systems, 2009. ISCAS 2009*, pp. 1917–1920.

Nakamura, J. (2006). *Image Sensors and Signal Processing for Digital Still Cameras*. English. Boca Raton, Fla. [u.a.: Taylor & Francis.

Neil, D. and S.-C. Liu (2014). "Minitaur, an Event-Driven FPGA-Based Spiking Network Accelerator". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 22.12, pp. 2621–2628.

Ni, Z. "Asynchronous Event Based Vision: Algorithms and Applications to Microrobotics". PhD thesis. url: `http://hal.inria.fr/docs/00/91/69/95/PDF/Thesis.pdf` (visited on 04/29/2014).

Ni, Z., A. Bolopion, J. Agnus, R. Benosman, and S. Regnier (2012). "Asynchronous Event-Based Visual Shape Tracking for Stable Haptic Feedback in Microrobotics". In: *IEEE Transactions on Robotics* 28.5, pp. 1081–1089.

Ni, Z., C. Pacoret, R. Benosman, and S. Regnier (2013). "2D high speed force feedback teleoperation of optical tweezers". In: *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1700–1705.

Nistér, D. (2005). "Preemptive RANSAC for live structure and motion estimation". en. In: *Machine Vision and Applications* 16.5, pp. 321–329. url: `http://link.springer.com/article/10.1007/s00138-005-0006-y` (visited on 11/16/2014).

O'Connor, P., D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer (2013). "Real-time classification and sensor fusion with a spiking deep belief network". In: *Frontiers in Neuroscience* 7. url: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3792559/` (visited on 04/29/2014).

Olsson, J. A. M. and P. Hafliger (2008a). "Mismatch reduction with relative reset in integrate-and-fire photo-pixel array". In: *IEEE Biomedical Circuits and Systems Conference, 2008. BioCAS 2008*, pp. 277–280.

Olsson, J. A. M. and P. Hafliger (2008b). "Two color asynchronous event photo pixel". In: *IEEE International Symposium on Circuits and Systems, 2008. ISCAS 2008*, pp. 2146–2149.

Olsson, J. A. M. and P. Hafliger (2009). "Live demonstration of an asynchronous integrate-and-fire pixel-event vision sensor". In: *IEEE International Symposium on Circuits and Systems, 2009. ISCAS 2009*, pp. 774–774.

Orchard, G., R. Benosman, R. Etienne-Cummings, and N. Thakor (2013). "A spiking neural network architecture for visual motion estimation". In: *2013 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 298–301.

Orchard, G., D. Matolin, X. Lagorce, R. Benosman, and C. Posch (2014a). "Accelerated frame-free time-encoded multi-step imaging". In: *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2644–2647.

Orchard, G. and R. Etienne-Cummings (2014b). "Bioinspired Visual Motion Estimation". In: *Proceedings of the IEEE* 102.10, pp. 1520–1536.

Orchard, G., J. Zhang, Y. Suo, M. Dao, D. T. Nguyen, S. Chin, C. Posch, T. D. Tran, and R. Etienne-Cummings (2012). "Real Time Compressive Sensing Video Reconstruction in Hardware". In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 2.3, pp. 604–615.

Paz-Vicente, R., A. Linares-Barranco, A. Jimenez-Fernandez, G. Jimenez-Moreno, and A. Civit-Balcells (2009). "Synthetic retina for AER systems development". In: *IEEE/ACS International Conference on Computer Systems and Applications, 2009. AICCSA 2009*, pp. 907–912.

Perez-Carrasco, J., T. Serrano-Gotarredona, C. Serrano-Gotarredona, B. Acha, and B. Linares-Barranco (2008). "High-speed character recognition system based on a complex hierarchical AER architecture". In: *IEEE International Symposium on Circuits and Systems, 2008. ISCAS 2008*, pp. 2150–2153.

Perez-Carrasco, J., B. Acha, C. Serrano, L. Camunas-Mesa, T. Serrano-Gotarredona, and B. Linares-Barranco (2010). "Fast Vision Through Frameless Event-Based Sensing and Convolutional Processing: Application to Texture Recognition". In: *IEEE Transactions on Neural Networks* 21.4, pp. 609–620.

Perez-Carrasco, J., B. Zhao, C. Serrano, B. Acha, T. Serrano-Gotarredona, S. Chen, and B. Linares-Barranco (2013). "Mapping from Frame-Driven to Frame-Free Event-Driven Vision Systems by Low-Rate Rate Coding and Coincidence Processing–Application to Feedforward ConvNets". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.11, pp. 2706–2719.

Perez-Pena, F., A. Morgado-Estevez, R. J. Montero-Gonzalez, A. Linares-Barranco, and G. Jimenez-Moreno (2011). "Video surveillance at an industrial environment using an address event vision sensor: Comparative between two different video sensor based on a bioinspired retina". In: *2011 Proceedings of the International Conference on Signal Processing and Multimedia Applications (SIGMAP)*, pp. 1–4.

Piatkowska, E., A. N. Belbachir, and M. Gelautz (2014). "Cooperative and asynchronous stereo vision for dynamic vision sensors". en. In: *Measurement Science and Technology* 25.5, p. 055108. url: `http://iopscience.iop.org/0957-0233/25/5/055108` (visited on 04/29/2014).

Picaud, S. and J.-A. Sahel (2014). "Retinal prostheses: Clinical results and future challenges". In: *Comptes Rendus Biologies*. Spotlight on vision Guest editor-in-chief / Rédacteur en chef invité : José-Alain Sahel 337.3, pp. 214–222. url: `http://www.sciencedirect.com/science/article/pii/S163106911400002X` (visited on 04/29/2014).

Posch, C., M. Hofstatter, D. Matolin, G. Vanstraelen, P. Schoˆn, N. Donath, and M. Litzenberger (2007a). "A Dual-Line Optical Transient Sensor with On-Chip Precision Time-Stamp Generation". In: *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, pp. 500–618.

Posch, C., M. Hofstatter, M. Litzenberger, D. Matolin, N. Donath, P. Schon, and H. Garn (2007b). "Wide dynamic range, high-speed machine vision with a 2x256 pixel temporal contrast vision sensor". In: *IEEE International Symposium on Circuits and Systems, 2007. ISCAS 2007*, pp. 1196–1199.

Posch, C., D. Matolin, and R. Wohlgenannt (2008a). "An asynchronous time-based image sensor". In: *IEEE International Symposium on Circuits and Systems, 2008. ISCAS 2008*, pp. 2130–2133.

Posch, C., D. Matolin, R. Wohlgenannt, T. Maier, and M. Litzenberger (2009). "A Microbolometer Asynchronous Dynamic Vision Sensor for LWIR". In: *IEEE Sensors Journal* 9.6, pp. 654–664.

Posch, C., D. Matolin, R. Wohlgenannt, M. Hofstatter, P. Scho?n, M. Litzenberger, D. Bauer, and H. Garn (2010a). "Biomimetic frame-free HDR camera with event-driven PWM image/video sensor and full-custom address-event processor". In: *2010 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pp. 254–257.

Posch, C., D. Matolin, and R. Wohlgenannt (2010b). "High-DR frame-free PWM imaging with asynchronous AER intensity encoding and focal-plane temporal redundancy suppression". In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2430–2433.

Posch, C. and D. Matolin (2011a). "Sensitivity and uniformity of a 0.18um CMOS temporal contrast pixel array". In: *2011 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1572–1575.

Posch, C., T. Serrano-Gotarredona, B. Linares-Barranco, and T. Delbruck (2014). "Retinomorphic Event-Based Vision Sensors: Bioinspired Cameras With Spiking Output". In: *Proceedings of the IEEE* 102.10, pp. 1470–1484.

Posch, C., R. Wohlgenannt, D. Matolin, and T. Maier (2008b). "A temporal contrast IR vision sensor". In: vol. 7100, 71002A–71002A–11. url: `http://dx.doi.org/10.1117/12.797700` (visited on 04/29/2014).

Posch, C., D. Matolin, and R. Wohlgenannt (2010c). "A two-stage capacitive-feedback differencing amplifier for temporal contrast IR sensors". en. In: *Analog Integrated Circuits and Signal Processing* 64.1, pp. 45–54. url: `http://link.springer.com/article/10.1007/s10470-009-9354-2` (visited on 04/29/2014).

Posch, C., D. Matolin, and R. Wohlgenannt (2011b). "A QVGA 143 dB Dynamic Range Frame-Free PWM Image Sensor With Lossless Pixel-Level Video Compression and Time-Domain CDS". In: *IEEE Journal of Solid-State Circuits* 46.1, pp. 259–275. url: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5648367` (visited on 08/13/2013).

Purves, D., G. J. Augistine, D. Fitzpatrick, L. C. Katz, A.-S. LaMantia, J. O. McNamara, and S. M. Williams (2001). *Neuroscience*. 2nd. Sinauer Associates.

Rea, F., G. Metta, and C. Bartolozzi (2013). "Event-driven visual attention for the humanoid robot iCub". In: *Frontiers in Neuroscience* 7. url: `http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3862023/` (visited on 04/29/2014).

Ríos, A., C. Conde, I. M. d. Diego, and E. Cabello (2014). "Driver's Hand Detection and Tracking Based on Address Event Representation". en. In: *Computational Modeling of Objects Presented in Images*. Ed. by P. D. Giamberardino, D. Iacoviello, R. N. Jorge, and J. M. R. S. Tavares. Lecture Notes in Computational Vision and Biomechanics 15. Springer International Publishing, pp. 131–144. url: `http://link.springer.com/chapter/10.1007/978-3-319-04039-4_8` (visited on 04/23/2014).

Ritz, R. (2008). "Development of a Dynamical Model for Retina-based Control of an RC Monster Truck". Bachelor Thesis. Zurich, Switzerland: ETH Zurich.

Rivas-Perez, M., A. Linares-Barranco, J. Cerda, N. Ferrando, G. Jime?nez, and A. Civit (2010). "Visual spike-based convolution processing with a Cellular Automata architecture". In: *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7.

Rocha, L., L. Velho, and P. Carvalho (2002). "Image moments-based structuring and tracking of objects". In: *XV Brazilian Symposium on Computer Graphics and Image Processing, 2002. Proceedings*, pp. 99–105.

Roclin, D., O. Bichler, C. Gamrat, S. Thorpe, and J.-O. Klein (2013). "Design study of efficient digital order-based STDP neuron implementations for extracting temporal features". In: *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7.

Rodgers, D. P. (1985). "Improvements in Multiprocessor System Design". In: *Proceedings of the 12th Annual International Symposium on Computer Architecture*. ISCA '85. Los Alamitos, CA, USA: IEEE Computer Society Press, pp. 225–231. url: `http://dl.acm.org/citation.cfm?id=327010.327215` (visited on 12/14/2014).

Rodieck, R. W. and M. Watanabe (1993). "Survey of the morphology of macaque retinal ganglion cells that project to the pretectum, superior colliculus, and parvicellular laminae of the lateral geniculate nucleus". en. In: *The Journal of Comparative Neurology* 338.2, pp. 289–303. url: `http://onlinelibrary.wiley.com/doi/10.1002/cne.903380211/abstract` (visited on 12/13/2014).

Rogister, P., R. Benosman, S.-H. Ieng, P. Lichtsteiner, and T. Delbruck (2012). "Asynchronous Event-Based Binocular Stereo Matching". In: *IEEE Transactions on Neural Networks and Learning Systems* 23.2, pp. 347–353.

Ros, P. M. and E. Pasero (2011). "Interfacing an AER Vision Sensor with a Tactile Display: a Proposal for a New Haptic Feedback Device". In: *Neural Nets WIRN11 - Proceedings of the 21st Italian Workshop on Neural Nets, Vietri sul Mare, Salerno, Italy, June 3-5, 2011*. Ed. by B. Apolloni, S. Bassis, A. Esposito, and F. C. Morabito. Vol. 234. Frontiers in Artificial Intelligence and Applications. IOS Press, pp. 332–343.

Roska, B. and F. Werblin (2001). "Vertical interactions across ten parallel, stacked representations in the mammalian retina". en. In: *Nature* 410.6828, pp. 583–587. url: `http://www.nature.com/nature/journal/v410/n6828/abs/410583a0.html` (visited on 08/18/2014).

Rosten, E. and T. Drummond (2005). "Fusing points and lines for high performance tracking". In: *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005*. Vol. 2, 1508–1515 Vol. 2.

Rublee, E., V. Rabaud, K. Konolige, and G. Bradski (2011). "ORB: An efficient alternative to SIFT or SURF". In: *2011 IEEE International Conference on Computer Vision (ICCV)*, pp. 2564–2571.

Ruedi, P.-F., P. Heim, F. Kaess, E. Grenet, F. Heitger, P.-Y. Burgi, S. Gyger, and P. Nussbaum (2003). "A 128 x 128 pixel 120-dB dynamic-range vision-sensor

chip for image contrast and orientation extraction". In: *IEEE Journal of Solid-State Circuits* 38.12, pp. 2325–2333.

SIFT: Introduction - AI Shack. url: `http://www.aishack.in/tutorials/sift-scale-invariant-feature-transform-introduction/` (visited on 12/14/2014).

Sarpeshkar, R., J. Kramer, G. Indiveri, and C. Koch (1996). "Analog VLSI architectures for motion processing: from fundamental limits to system applications". In: *Proceedings of the IEEE* 84.7, pp. 969–987.

Sarpeshkar, R. (1997). "Efficient precise computation with noisy components : extrapolating from an electronic cochlea to the brain." en. PhD thesis. Sacramento, CA, USA: California Institute of Technology. url: `http://thesis.library.caltech.edu/3063/`.

Schoitsch, E., C. Sulzbachner, and J. Kogler (2013). "Enhanced Low-Cost Sensing Technologies for Vehicle On-Board Safety Applications (ADOSE Project)". en. In: *Advanced Microsystems for Automotive Applications 2013*. Ed. by J. Fischer-Wolfarth and G. Meyer. Lecture Notes in Mobility. Springer International Publishing, pp. 111–121. url: `http://link.springer.com/chapter/10.1007/978-3-319-00476-1_11` (visited on 04/29/2014).

Schrag, M. (2008). "Realtime Topology Learning". Semester Thesis. Zur: ETH Zurich. url: `www.ini.uzh.ch/~tobi/wiki/lib/exe/fetch.php?media=schragrealtimetopologylearning2008.pdf`.

Schraml, S., A. Belbachir, and N. Brandle (2010a). "A real-time pedestrian classification method for event-based dynamic stereo vision". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 93–99.

Schraml, S. and A. Belbachir (2010b). "A spatio-temporal clustering method using real-time motion analysis on event-based 3D vision". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 57–63.

Schraml, S., A. Belbachir, N. Milosevic, and P. Scho?n (2010c). "Dynamic stereo vision system for real-time tracking". In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1409–1412.

Schraml, S., A. Belbachir, N. Milosevic, and P. Scho?n (2010d). "Live demonstration: Dynamic stereo vision system for real-time tracking". In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1408–1408.

Serrano-Gotarredona, R., M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz, F. Gómez-rodríguez, H. K. Riis, T. Delbrück, S. C. Liu, S. Zahnd, A. M. Whatley, R. Douglas, P. Häfliger, G. Jimenez-moreno, and A. Civit (2005). "AER building blocks for multi-layer multi-chip neuromorphic vision systems". In: *in Advances in Neural Information Processing Systems*. MIT Press, pp. 1217–1224.

Serrano-Gotarredona, R., T. Serrano-Gotarredona, A. Acosta-Jimenez, and B. Linares-Barranco (2006). "An arbitrary kernel convolution AER-transceiver chip for real-time image filtering". In: *2006 IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006. Proceedings*, 4 pp.–.

Serrano-Gotarredona, R., T. Serrano-Gotarredona, A. Acosta-Jimenez, C. Serrano-Gotarredona, J. Perez-Carrasco, B. Linares-Barranco, A. Linares-Barranco, G. Jimenez-Moreno, and A. Civit-Ballcels (2008). "On Real-Time AER 2-D Convolutions Hardware for Neuromorphic Spike-Based Cortical Processing". In: *IEEE Transactions on Neural Networks* 19.7, pp. 1196–1219.

Serrano-Gotarredona, R., M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gomez-Rodriguez, L. Camunas-Mesa, R. Berner, M. Rivas-Perez, T. Delbruck, S.-C. Liu, R. Douglas, P. Hafliger, G. Jimenez-Moreno, A. Ballcels, T. Serrano-Gotarredona, A. Acosta-Jimenez, and B. Linares-Barranco (2009). "CAVIAR: A 45k Neuron, 5M Synapse, 12G Connects/s AER Hardware Sensory #x2013;Processing #x2013; Learning #x2013;Actuating System for High-Speed Visual Object Recognition and Tracking". In: *IEEE Transactions on Neural Networks* 20.9, pp. 1417–1438.

Serrano-Gotarredona, T., J. Park, A. Linares-Barranco, A. Jimenez, R. Benosman, and B. Linares-Barranco (2013a). "Improved contrast sensitivity DVS and its application to event-driven stereo vision". In: *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2420–2423.

Serrano-Gotarredona, T. and B. Linares-Barranco (2013b). "A 128 x 128 1.5% Contrast Sensitivity 0.9% FPN 3 μs Latency 4 mW Asynchronous Frame-Free Dynamic Vision Sensor Using Transimpedance Preamplifiers". In: *IEEE Journal of Solid-State Circuits* 48.3, pp. 827–838. url: `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6407468` (visited on 08/13/2013).

Shoushun, C. and A Bermak (2007). "Arbitrated Time-to-First Spike CMOS Image Sensor With On-Chip Histogram Equalization". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 15.3, pp. 346–357.

Silveira, R. Azeredo da and B. Roska (2011). "Cell Types, Circuits, Computation". In: *Current Opinion in Neurobiology*. Networks, circuits and computation 21.5, pp. 664–671. url: `http://www.sciencedirect.com/science/article/pii/S0959438811000754` (visited on 08/18/2014).

Sivilotti, M. A. (1991). "Wiring considerations in analog VLSI systems, with application to field-programmable networks". phd. California Institute of Technology. url: `http://resolver.caltech.edu/CaltechETD:etd-07122007-134330` (visited on 09/11/2014).

Smirnakis, S. M., M. J. Berry, D. K. Warland, W. Bialek, and M. Meister (1997). "Adaptation of retinal processing to image contrast and spatial scale". en. In: *Nature* 386.6620, pp. 69–73. url: `http://www.nature.com/nature/journal/v386/n6620/abs/386069a0.html` (visited on 05/19/2014).

Sonnleithner, D. and G. Indiveri (2011). "A neuromorphic saliency-map based active vision system". In: *2011 45th Annual Conference on Information Sciences and Systems (CISS)*, pp. 1–6.

Sonnleithner, D. and G. Indiveri (2012). "A Real-Time Event-Based Selective Attention System for Active Vision". In: *Advances in Autonomous Mini Robots*. Ed. by U. Rückert, S. Joaquin, and W. Felix. Springer Berlin Heidelberg, pp. 205–219. url: `http://link.springer.com/chapter/10.1007/978-3-642-27482-4_21` (visited on 04/29/2014).

Spiegel, J. Van der (1996). "Computational sensors - the basis for truly intelligent machines". In: *Intelligent Sensors, Elsevier, Amsterdam*, pp. 19–38.

Stocker, A. (2004). "Analog VLSI focal-plane array with dynamic connections for the estimation of piecewise-smooth optical flow". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 51.5, pp. 963–973.

Strubel, J. (2013a). "Knee tracking for control of active exoprostheses using a Dynamic Vision Sensor". en. Master Thesis. Zurich, Switzerland: University of Zurich. url: http://www.merlin.uzh.ch/publication/show/8459.

Strubel, J. (2013b). "Quadrotor tracking and pose estimation using a dynamic vision sensor". Semester Thesis. Zurich, Switzerland: University of Zurich.

Sulzbachner, C., M. Humenberger, Á. Srp, and F. Vajda (2012). "Optimization of a Neural Network for Computer Vision Based Fall Detection with Fixed-Point Arithmetic". In: *Neural Information Processing*. Ed. by T. Huang, Z. Zeng, C. Li, and C. S. Leung. Lecture Notes in Computer Science 7666. Springer Berlin Heidelberg, pp. 18–26. url: http://link.springer.com/chapter/10.1007/978-3-642-34478-7_3 (visited on 04/29/2014).

Tanner, J. E. (1986). "Integrated optical motion detection". phd. California Institute of Technology. url: http://resolver.caltech.edu/CaltechETD:etd-03102008-081506 (visited on 05/20/2014).

Teifel, J. and R. Manohar (2004). "An asynchronous dataflow FPGA architecture". In: *IEEE Transactions on Computers* 53.11, pp. 1376–1392.

Thomson, S. W. (1881). "THE TIDE GAUGE, TIDAL HARMONIC ANALYSER, AND TIDE PREDICTER." en. In: *Minutes of the Proceedings* 65.1881, pp. 2–25. url: http://www.icevirtuallibrary.com/content/article/10.1680/imotp.1881.22262 (visited on 12/13/2014).

Thoreson, W. B. and S. C. Mangel (2012). "Lateral interactions in the outer retina". In: *Progress in Retinal and Eye Research* 31.5, pp. 407–441.

Thorpe, S., A. Brilhault, and J.-A. Perez-Carrasco (2010). "Suggestions for a biologically inspired spiking retina using order-based coding". In: *Proceedings of 2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 265–268.

Thurnherr, M. (2012). "Prosthesis With Eyes - A Feasibility Study". en. Bachelor Thesis. Zurich, Switzerland: ETH Zurich.

Tschechne, S., R. Sailer, and H. Neumann (2014). "Bio-Inspired Optic Flow from Event-Based Neuromorphic Sensor Input". en. In: *Artificial Neural Networks in Pattern Recognition*. Ed. by N. E. Gayar, F. Schwenker, and C. Suen. Lecture Notes in Computer Science 8774. Springer International Publishing, pp. 171–182. url: http://link.springer.com/chapter/10.1007/978-3-319-11656-3_16 (visited on 10/05/2014).

Tureczek, A. (2008). "Active shape models on Silicon retina". Master Thesis. Zurich, Switzerland: ETH Zurich. url: www.ini.uzh.ch/~tobi/wiki/lib/exe/fetch.php?media=tureczekfacetrackingdvs2008.pdf.

Venn, J. (2007). *Symbolic Logic*. English. Kessinger Publishing, LLC.

Viola, P. and M. Jones (2001). "Rapid object detection using a boosted cascade of simple features". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001*. Vol. 1, I–511–I–518 vol.1.

Vogelstein, R. J., U. Mallik, E. Culurciello, G. Cauwenberghs, and R. Etienne-Cummings (2007). "A Multichip Neuromorphic System for Spike-Based Visual Information Processing". In: *Neural Computation* 19.9, pp. 2281–2300. url: `http://dx.doi.org/10.1162/neco.2007.19.9.2281` (visited on 04/24/2014).

Von Neumann, J., S. N. Alexander, f. o. D. United States. National Bureau of Standards, United States. Army. Ordnance Dept, and University of Pennsylvania (1945). *First draft of a report on the EDVAC*. eng. Philadelphia : Moore School of Electrical Engineering, University of Pennsylvania. url: `http://archive.org/details/firstdraftofrepo00vonn` (visited on 12/13/2014).

Weber, A. S. (2000). *Nineteenth-Century Science: An Anthology*. en. Broadview Press.

Weikersdorfer, D., R. Hoffmann, and J. Conradt (2013). "Simultaneous Localization and Mapping for Event-Based Vision Systems". In: *Computer Vision Systems*. Ed. by M. Chen, B. Leibe, and B. Neumann. Lecture Notes in Computer Science 7963. Springer Berlin Heidelberg, pp. 133–142. url: `http://link.springer.com/chapter/10.1007/978-3-642-39402-7_14` (visited on 04/29/2014).

Wiesmann, G., S. Schraml, M. Litzenberger, A. N. Belbachir, M. Hofstatter, and C. Bartolozzi (2012). "Event-Driven Embodied System for Feature Extraction and Object Recognition in Robotic Applications". In: *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 76–82.

Wikipedia. en. Free Encyclopedia (public domain image). url: `http://en.wikipedia.org/`.

Wohlgenannt, R., D. Matolin, T. Maier, and C. Posch (2008). "Characterization of a temporal contrast microbolometer infrared sensor". In: *15th IEEE International Conference on Electronics, Circuits and Systems, 2008. ICECS 2008*, pp. 866–869.

Yang, M., S.-C. Liu, C. Li, and T. Delbruck (2012). "Addressable Current Reference Array with 170dB Dynamic Range". In: *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 3110–3113.

Zaghloul, K. A. and K. Boahen (2004a). "Optic nerve signals in a neuromorphic chip I: Outer and inner retina models". eng. In: *IEEE transactions on bio-medical engineering* 51.4, pp. 657–666.

Zaghloul, K. A. and K. Boahen (2004b). "Optic nerve signals in a neuromorphic chip II: Testing and results". eng. In: *IEEE transactions on bio-medical engineering* 51.4, pp. 667–675.

Zaghloul, K. A. and K. Boahen (2006). "A silicon retina that reproduces signals in the optic nerve". en. In: *Journal of Neural Engineering* 3.4, p. 257. url: `http://iopscience.iop.org/1741-2552/3/4/002` (visited on 09/13/2014).

Zhang, X. and S. Chen (2012). "A hybrid-readout and dynamic-resolution motion detection image sensor for object tracking". In: *2012 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1628–1631.

Zhang, X. and S. Chen (2013). "Live demonstration: A high-speed-pass asynchronous motion detection sensor". In: *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 671–671.

Zhao, B., S. Chen, and H. Tang (2014). "Bio-inspired categorization using event-driven feature extraction and spike-based learning". In: *2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 3845–3852.

Zuse, H. "Anmerkungen zum John von Neumann Rechner". url: `http://www.horst-zuse.homepage.t-online.de/fiff99-a-2006.pdf`.

Zuse, K., F. L. Bauer, and H. Zemanek (2010). *The Computer - My Life*. English. Trans. by P. McKenna and J. A. Ross. Softcover reprint of hardcover 1st ed. 1993 edition. Berlin, Heidelberg: Springer.

jAER. url: `https://sourceforge.net/p/jaer/wiki/Home/` (visited on 09/17/2013).