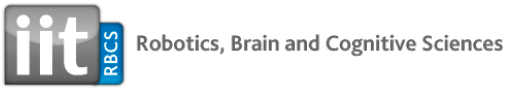
Implementation of Neuromorphic Vision Solutions for Humanoid Robots



Charles Clercq

Robotics, Brain and Cognitive Sciences

Fondazione Istituto Italiano di Tecnologia

in co-tutoring with

Institut de la Vision

A thesis submitted to the Department of *SISTEMI DI ELABORAZIONE DELLE INFORMAZIONI* in fulfillment of the requirements for the degree of *Doctor in*

ROBOTICS, COGNITION AND INTERACTION TECHNOLOGIES

2013 April

Thesis Supervisors:

Chiara Bartolozzi
Team leader
Department of Robotics, Brain and Cognitive Sciences
Fondazione Istituto Italiano di Tecnologia

Ryad Benosman
Professor
Department of Natural Computation
Institut de la Vision

Abstract

Artificial vision aims to confer to machines the ability to perceive and to interpret their environment by inspiring from the biological vision. Since the beginning of image processing and machine vision fields in the 60's, numerous techniques have been developed to extract and to process visual information in an all implicitly accepted and unique context of frames. This is shown to be in contradiction with biologic eyes which have discarded through evolution mechanisms the concepts of frame and synchronized pixels. This thesis aims to switch the classic computer vision paradigm based on frame for the biological one. New vision algorithms are developed using this new paradigm for the iCub humanoid robot, in the context of the european project eMorph. The aimed goal is to design a neuro-inspired vision based navigation ability to the robot. This neuro-inspired vision is expected to be much more accurate and energy efficient.

Dedication

A Mancie.

Acknowledgements

I would like to thank all the people who helped me during these three years of PhD.

Alice, Alix, Beatrice, Cedric, Elena, Emma, Emilie, Fouzhan, Francesco, Henri, Joao, Justine, Marco, Marianna, Martina, Mitzu, Neeraj, Sio, Xavier, Xavier, and my supervisors.

A special thank to Michael H. who makes me understand that, apart from the precise timing, nothing is a matter of time, but of priorities.

I would also like to thank the Neuromorphic Community and the organisators of the *CapoCaccia Cognitive Neuromorphic Engineering Workshop*.

Contents

1 Introduction **1**

1.1 eMorph European Project 1

1.1.1 Limitations of the current computer vision paradigm 3

1.1.2 Toward an asynchronous event based vision paradigm 3

1.1.3 Asynchronous vision sensors : state of the art 6

1.1.4 Asynchronous Vision Algorithms: state of the art 9

1.2 Robotic navigation 10

1.2.1 State of the art 10

1.2.2 Event-based Navigation 12

2 Methods **15**

2.1 Event-based formalism 15

2.1.1 Pre-requisite 15

2.1.1.1 From luminance to temporal changes 15

2.1.1.2 Codifications strategies to encode light changes . . . 18

2.1.2 Properties of event-driven acquisition 21

2.1.2.1 Decomposition of spatiotemporal volumes 21

2.1.2.2 Noise 22

2.1.2.3 Events and images 23

2.2 Event-based computation 24

2.2.1 Temporal convolution 25

2.2.2 Spatial convolution 26

3 Event based visual navigation **27**

3.1 The optical flow 27

3.1.1 State of the art 27

3.1.2 Asynchronous frameless event-based flow 29
3.1.2.1 Assumption and equations of the optical flow 29
3.1.3 Event-based visual flow 32
3.1.3.1 Flow definition 32
3.1.3.2 Flow regularization 33
3.2 Focus of expansion 36
3.2.1 Definition 36
3.2.2 State of the art 36
3.2.3 Equations 37
3.3 Time to contact 38
3.3.1 Definition 38
3.3.2 State of the art 39
3.3.3 Equations 40

4 Results and validation **43**
4.1 The optical flow 43
4.1.1 Method 1: Asynchronous frameless event-based optical flow . 44
4.1.2 Method 2: Event based motion flow 49
4.1.3 Limitations 57
4.2 Focus of expansion and Time to contact 63
4.3 Focus of expansion 63
4.3.1 Simulated data 66
4.3.2 Real data 69
4.4 Time to contact 70

5 Conclusion **72**
5.1 Asynchronous frameless event-based flow 72
5.2 Visual flow 73
5.3 Focus of expansion and Time to contact 74

6 Discussion **76**

References **78**

Chapter 1

Introduction

1.1 eMorph European Project

Mainstream computational paradigm in embodied intelligence is digital and it is clear that conventional digital systems have difficulties in performing robustly even in the most mundane tasks of perception. They require vast amounts of resources to extract relevant information, but still fail to produce appropriate responses for interacting with the real-world in real time. In addition, in sensory perception tasks, the data acquired from the sensors are typically noisy and ambiguous. Frame-based time sampling and quantization artifacts present in conventional sensors are particularly problematic for robust and reliable performance.

The situation is clearly different in biological systems. In particular, biological neural systems vastly outperform conventional digital machines in almost all aspects of sensory perception tasks. Despite its dramatic progress, information technology has not yet been able to deliver artificial systems that can compare with biology. There are limitations both at the technological level, and at the theoretical/computational level.

Analog computation free from the limits of sampling provides a solution. Analog

devices are fast, as time constants are in the range of the rising time of the transistor currents. Event-driven computation intrinsically adapts the sensor response to the time constants of the real world. The sensor response is automatically regulated to match the incoming signal range, and so is robust. Moreover as only important events are coded, they are also efficient. The eMorph project thus indented to design novel, data-driven, biologically inspired, analog sensory devices while also developing new asynchronous event-driven computational paradigms for them.

eMorph aimed to adapt the computational engine of the cognitive system to the dynamics of the real world rather than furiously sample the physical sensory signals in an attempt to obtain adequate bandwidth. Structure and morphology have been matched to the requirements of the robots body and its application domain with testing carried out on the advanced humanoid robotic platform, iCub (project RobotCub).

The general objectives of eMorph were to implement embodied intelligence by designing space-variant morphologies and computational structures of neuromorphic sensors together with the development of asynchronous data-driven algorithms that best exploit the properties of the sensor.

My research focused on the specific problem of motion perception and computation of optical flow, in the context of obstacle avoidance during navigation. The resulting algorithms have been developed and tested in controlled situations and eventually implemented and validated on the iCub robot.

In the following, I will highlight the limitations of mainstream approaches in computer vision in general and then present the state of the art of the asynchronous visual sensors and relative sensory systems. Afterwards I will give an overview of the robotic navigation, and then explain why the event-based paradigm is crucial for this task.

1.1.1 Limitations of the current computer vision paradigm

The field of computer vision is seen by other robotics disciplines as immature and still too diverse. Even though earlier work exists, only when computers could manage the processing of large data sets such as images, in the late 1970s a more focused study of the field started. However, these studies usually originated from various other fields, and consequently there is no standard formulation of "the computer vision problem".

There is no standard formulation of how computer vision problems should be solved. Instead, there exists an abundance of methods for solving various well-defined computer vision tasks, where the approaches are often very task specific and cannot be generalized over a wide range of applications.

Many of the methods and applications are still in the state of basic research, but more and more find their way into commercial products, where they often constitute part of a larger system which can solve complex tasks (e.g., in the area of medical imaging, or quality control and monitoring of industrial processes).

In most computer vision applications, systems are pre-programmed to solve particular tasks, but methods based on learning are now becoming increasingly common.

1.1.2 Toward an asynchronous event based vision paradigm

The notion of a "frame" of video data has become so embedded in machine vision that it is usually taken for granted. This is natural given that frame-based devices have been dominant from the days of drum scanners and videcon tubes to todays CCDs and CMOS imagers. There are undeniable advantages to frame-based imagers: they have small simple pixels, leading to high resolution, large fill factor and low imager cost. The output format is well understood and is the basis for many years of research in machine vision.

On the other hand, frame-based architectures carry hidden costs because they

are based on series of snapshots taken at a constant rate, irrespective of the scene content, thereby pixels are repetitively sampled even if their values are unchanged. Short-latency vision problems require high frame rate and produce massive output data. Pixel bandwidth is limited to half of the frame rate, and reducing the output to a manageable rate by using region-of-interest readout usually requires complex control strategies. Dynamic range is typically limited by the identical pixel gain, the finite pixel capacity for integrated photocharge, and the identical integration time. For machine vision in uncontrolled environments with natural lighting, limited dynamic range and bandwidth can compromise performance.

The classical problem in computer vision, image processing and machine vision is that of determining whether or not the image data contains some specific object, feature, or activity. This task can normally be solved robustly and without effort by a human, but is still not satisfactorily solved in computer vision for the general case: arbitrary objects in arbitrary situations. The existing methods for dealing with this problem can at best solve it only for specific objects, such as simple geometric objects (e.g., polyhedrons), human faces, printed or hand-written characters, or vehicles, and in specific situations, typically described in terms of well-defined illumination, background, and pose of the object relative to the camera.

Real-world robotics applications are evolving from the industrial domain (simple tasks in structured environment) to the service domain where the environment of the robot is no more static. Service robotics induces complexity both in terms of the tasks that have to be achieved and in terms of the nature of the environment where robots are supposed to evolve. Part of the answer to these problems is the growing complexity of the sensors with which robots are now equipped as well as the increase of the number of degrees of freedom of the robots themselves e.g., Mobile manipulators such as the humanoid robot iCub (1) or the wheeled assistant PR2 (2)).

As a matter of fact, robots need a real-time representation of the dynamic environment that is robust to changing and uneven illumination and noise. Motion controllers have to be either highly robust to uncertainties on the knowledge of the model of the robot and its environment, or adaptive.

Fig. 1.1 illustrates a global process of current artificial vision methods, based on the acquisition of discretized frames. The process is very long, and has an important

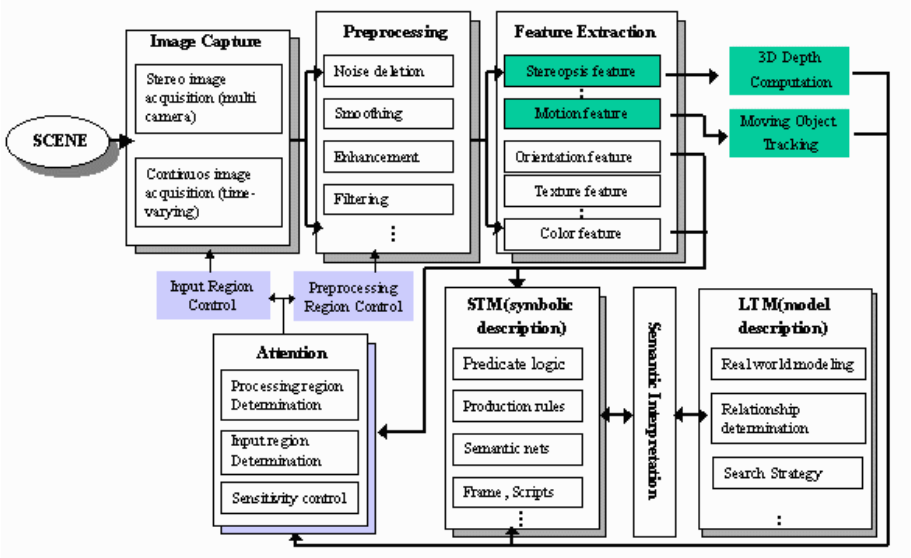


Figure 1.1: Global computer vision process

cost in terms of computation and memory communication bandwidth, specially for high frame-rate applications, such as motion computation, as proposed in this thesis.

Contrary to mainstream paradigms, biological systems are massively parallel and event driven. The retina is composed of multiple cells (rods and cones), each independently reacting to changes of light, as shown in Fig. 1.2 that illustrates the cells response to a spot light flash.

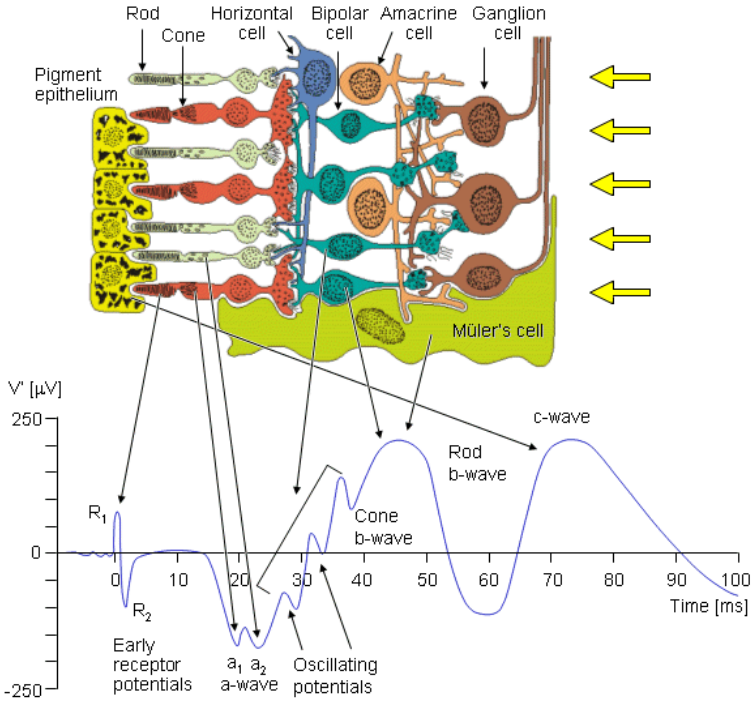


Figure 1.2: Illustration of the cells of the retina and their response to a spot light flash. The photo-receptors are the rods and cones in which a negative receptor potential is elicited. This drives the bipolar cell to become either depolarized or hyper-polarized. The amacrine cell has a negative feedback effect. The ganglion cell fires an action pulse so that the resulting spike train is proportional to the light stimulus level.

1.1.3 Asynchronous vision sensors : state of the art

The first asynchronous, data-driven and biologically inspired sensory device providing spike-based output was built by Mahowald and Mead (3) in the '80s, but was a demonstration device that was unusable for real world task.

Following, some other designs have been proposed: Zaghoul and Boahen (4) led to large mismatch, the pixel firing rates varied with a standard deviation of 1-2 decades and more than half the pixels did not spike at all for stimuli with 50% contrast making it impractical in real applications.

The group at CSEM Neuchatel (5) presented a device in which the output encodes spatial rather than temporal contrast. After a global frame integration period,

this device transmits events in the order of high-to-low spacial contrast. The main limitation of this architecture is the none reducing temporal redundancy, and the temporal resolution limited to the frame rate.

Etienne-Cumming's group reported a temporal change threshold detection imager (6), which modifies the traditional active pixel sensor CMOS so that it can detect a quantized absolute change in illumination. Frame-based, one of the disadvantage of this device is that the event times are quantized to the limited global sample rate.

Culurciello and Andreou (7) reported several imaging sensors that use Address-Event Representation (AER) to communicate the pixel intensity, either by inter-event interval or mean frequency. They have the advantage of relatively small pixel size, but the big disadvantage that the bus bandwidth is allocated according to the local scene luminance. Because there is no reset mechanism and because the event interval directly encodes intensity, a dark pixel can take a long time to emit an event, and a single highlight in the scene can saturate the bus.

Patrick Lichtsteiner *et al* (8) proposed the Dynamic Vision Sensor (DVS), a 128x128 pixel CMOS vision sensor, where each pixel independently generates spike events in continuous time quantizes local relative intensity as shown in Fig.1.3.

By providing high pixel bandwidth, wide dynamic range, and precisely timed sparse digital output, the DVS provides an attractive combination of characteristics for low-latency dynamic vision under uncontrolled illumination with low post-processing requirements. Pixels respond asynchronously to relative changes in intensity. The objective for this pixel design was to achieve low mismatch, wide dynamic range, and low latency in a reasonable pixel area.

The DVS has the potential of realization of small, fast, low power embedded sensory-motor processing systems that are beyond the reach of traditional approaches under the constraints of power, memory and processor cost. The field of

this kind of device, as AER sensor, is largely unexplored, and need improvements.

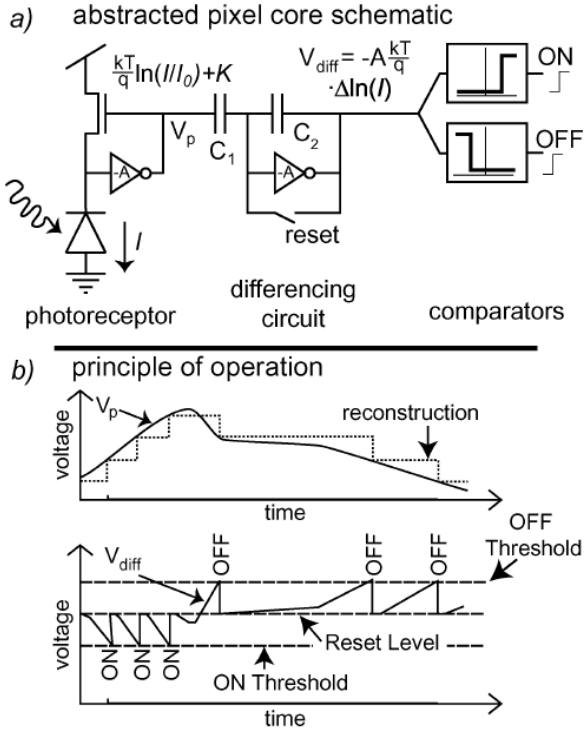


Figure 1.3: (a) Abstracted pixel schematic. (b) Principle of operation. In (a), the inverters are symbols for single-ended inverting amplifiers.

Posh et al at the Austrian Institute of Technology developed the Asynchronous Time-based Image Sensor (ATIS)(9). This device combines DVS pixels(8) with time-based photo-measurement pixels providing a PWM intensity encoding(10). The output is gray level values only at pixels which local luminosity changes. The ATIS made a step forward in AER functionality by providing typical local luminosity changes information coming with asynchronous real gray value measurements. The ATIS has a higher resolution than the DVS with 304x240 pixels, which is large given the youthfulness of the technology.

Linares-Barrancos group in Sevilla has presented an address event sensor consisting in computing the spatial contrast, coding it as pixel event frequency (rate

coding)(11). The chip also includes a global pixel reset mechanism which allows a Time-to-First-Spike coding.

Several algorithms have been developed using asynchronous sensors, mostly with DVS and ATIS. While applications using ATIS are mostly produced at the Austrian Institute of Technology (AIT) for commercial purpose (i.e. Fast moving object detection and classification, High-speed measurement tasks in industrial automation), DVS, by its availability and simplicity to use, is essentially employed for academic research.

Along this thesis I focus on the DVS starting by an overview of the state of the art involving this sensor.

1.1.4 Asynchronous Vision Algorithms: state of the art

The DVS is the first asynchronous sensor with low mismatch and user friendly programmable bias tuning that allows its use in artificial vision applications. Several techniques have been developed taking advantage of the low redundancy and high bandwidth of the sensor. In (13) the stream of events of the DVS is used to track ball using clusters. The clusters are incrementally updated and used to block balls with a motor-controlled arm. (14) takes advantage of the sensor to provide fast visual feedback for controlling an actuated table to balance an ordinary pencil. The pencil is tracked using a vertical shape, which position and angle are computed from two DVS output. (15) and (16) make use of the DVS in micro-manipulation and tracking. First technique tracks micro-spheres computing hough circles, while the second make use of an event-based Iterative Closest Point (ICP). In (17) the core interest of the information provided by the DVS is the events timing. Exploiting the visual event temporal occurrence of a pair of sensors, the proposed algorithm is able to accurately reconstruct the depth of moving objects.

The work developed so far shows that the DVS is suitable for applications where

fast stimuli have to be processed in real-time. To give autonomy to mobile robots, we chose to exploit the sensors for navigation by computing optical flow. The optical flow provides apparent motions, and is a basis of focus of expansion and time to contact computations as I will show in chapter3.

The event-driven approach has an enormous advantage in terms of temporal resolution, allowing for the computation of speed for fast moving objects, additionally, the intrinsically no-redundant data encoding lowers the computational demand for this usually power hungry algorithm.

Navigation is an important topic in the robotic field. In the following I give an overview of the state of the art and then interest points leading the choice of an event-based paradigm for robotic navigation.

1.2 Robotic navigation

The aim of the robotic navigation is to confer autonomy to the robot, giving the ability to navigate in an environment by its own. It consists of reaching a goal point using a suitable and safe path.

1.2.1 State of the art

Autonomous navigation in an unstructured environment is a very challenging task. Current robots that can operate autonomously in an unknown and unprepared environments are often huge as most of today's autonomous navigation algorithms rely on power-hungry sensors such as laser range finders and high resolution stereo-visions ((18)(19)(20)(21)(22)). These robots require then powerful and power-hungry computing units to be mounted on-board, leading to several limitations on autonomy and operational range.

Most obstacle avoidance algorithms use active range sensors such as ultrasonic

sensors, laser range finders and infrared sensors. Visual sensors are an alternative solution for obstacle avoidance and becoming increasingly popular in robotics. Visual sensors often provides better resolution data, longer ranges at faster rates than range sensors. Such visual sensors are passive, they are less dependent on the environment, however, image processing from commonly used frame-based cameras is a very computationally expensive task.

Concerning the vision based navigation, we can divide the different approaches in two main categories(23), the model based and the model-free.

The model based paradigm implies a partial or global knowledge of the environment before any action from the robot. Different strategies are used, consisting of different representations of the environment. The most complete representation is given by the metric map approach(24)(25) where the environment is totally known. To be used, the robot needs a system of localization like Simultaneous Localization and Mapping (SLAM) and Concurrent Mapping and Localization (CML). A less complete approach, but still using a global representation of the environment is based on the use of topological maps(26)(27), representing the environment like a graph, which nodes can represent actions. This approach does not represent obstacles, and implies the robot to be equipped with sensors to detect them. Topological maps are simple and compact, take up less computer memory, and consequently speed up the navigation processes. A transitional approach to model-free navigation is the local map method(28)(29). It consists in the creation of local map, which updates incrementally a more global representation of the environment. The local grid represents the portion of the environment that surrounds the robot, the grid size being determined by the sensor field of view.

The model-less methods of navigation are based on the fast reaction to obstacles and consequent path recalculation in an unstructured environment of reactive navigation, where robots have to deal with unstructured environment through his

motion. As the model based method, the model-less is subdivided in different approaches. Feature tracking uses the detection of particular shapes such as corners, lines, which can be used to segment ground plane(30). Another interesting approach, the Image Qualitative Characteristic Extraction, avoids as much as possible the use, computation or generation of accurate numerical data such as distances, position coordinates, velocities or contact time to obstacles(31)(32). This approach suffers from changing imaging conditions, i.e. illumination intensity, position of light sources, glossiness of the scene materials. A last subdivision of the model-free methods is represented by the computation of the optical flow. In robotic application, the optical flow is used for direct navigation as taking inspiration of insect(33) for instance. The information provided by the apparent motion allows to detect obstacles(34)(35) and also to compute the time before their impact(36).

Fast navigation is a fundamental problem in robotics. Current navigation methods, as quickly presented above, are designed to work on frames, usually acquired at a fixed rate using conventional cameras. The resulting data redundancy leads to high computational burden while the image acquisition temporal resolution, often limited to video frame rates of the order of tens of milliseconds, restricting the speed of mobile platforms.

Vision often requires complicated software and powerful computing platform or dedicated hardware modules. Very small robots, i.e. those that are man-transportable are not usually equipped with vision sensors. Crowlay underline in(37) that one of the most important problems to be addressed in visual navigation is sensor interpretation. This task is indubitably increased by the redundancy of standard imagers.

1.2.2 Event-based Navigation

A recent and evolving area in computer vision aims at exploiting the unique characteristic of a novel family of biologically inspired asynchronous sensors (38)(39)

(10)(40). The increasing availability and the quality of these sensors open up the potential to introduce a shift in the methodology of acquiring and processing visual information in various, demanding machine vision applications(41)(42). The sub-microsecond temporal resolution and the inherent redundancy suppression of the frame-free, event-driven acquisition and subsequent representation of visual information employed by these cameras enable to derive a novel methodology to process the visual information at an unprecedented speed and with low computational cost.

For these reasons, the use of this kind of sensor is promising for robot's navigation, for which we had to develop algorithms with new methodologies.

We focused on the optical flow approach because the nature of the sensors. As only dynamic part of a scene is provided, the optical flow was viewed as naturally emerging. Moreover, the importance of this work is emphasized by Gibson claim in(43), that the optical flow is used by humans to extract all the necessary environmental information, such as egomotion or obstacle's positions.

In this thesis I describe a novel computationally efficient and robust event-based chain of visual navigation that relies on the accurate timing of individual pixels' response allowing in a first step to compute an efficient and accurate optical flow. This method takes the visual stimuli as computational inputs and fully benefits of the high temporal resolution properties of the sensor. The second step of the navigation chain is the computation of the focus of expansion (FOE), that provides egomotion information. The joint use of the focus of expansion and the optical flow allows the computation of the time to contact (TTC), which provides the information of the time before impact of nearby obstacles.

In the following I present the work done to accomplish the task of safe navigation. Chapter2 describes the event-based paradigm, and give a general methodology to compute with spikes. Chapter 3 details two approaches of optical flow computation, the focus of expansion and the time to contact calculation. Eventually Chapter

4 presents the validation of the methods details in chapter 3. Finally chapter 5 concludes on the results of the presented methods and chapter 6 gives an overview for further work.

Chapter 2

Methods

In this chapter, I introduce the event-based paradigm, giving the mathematical foundation of the approach. First, I explain how event-based sensors discretize the luminance of a scene, and how this information is encoded. Then, I explain how events can be mathematically treated and the properties and computational capabilities of the developed algorithms.

2.1 Event-based formalism

2.1.1 Pre-requisite

2.1.1.1 From luminance to temporal changes

A gray-scale image is a $M \times N$ array of positive numbers encoding the intensity of light incident on the pixels located on the sensor image plane.

A movie is a sequence, I_{seq}, of images (or frames) periodically taken at a fixed temporal interval, Δt, from t_0 to $t_0 + n\Delta t$:

$$I_{seq} = \{I(t_0), I(t_0 + \Delta t), \ldots, I(t_0 + n\Delta t)\} \tag{2.1}$$

with $n \in \mathbb{N}^+$.

The intensity of a given a pixel at $[x, y]^t$, is a mono-dimensional function of time t:

$$\begin{array}{rccc} f_{x,y}: & [0,T] & \rightarrow & \mathbb{R}^+ \\ & t & \mapsto & f_{x,y}(t) \end{array} . \tag{2.2}$$

A digital image taken at time t is the set of mono-dimensional functions $f_{x,y}$:

$$I(t) = \{f_{x,y}(t)\}_{\substack{x \in [0, M-1] \\ y \in [0, N-1]}} . \tag{2.3}$$

Using Eq. 2.2, an image sequence (Eq. 2.1) can be written as a set of functions $f_{x,y}(t)$ expressing the variations of the values of pixels over time:

$$I_{seq} = \{f_{0,0}(t), f_{0,1}(t), \ldots, f_{M-1,N-1}(t)\} \tag{2.4}$$

for $t \in [t_0, t_0 + n\Delta t]$, and $n \in \mathbb{N}^+$.

I_{seq} is composed of a set of discrete functions $f_{x,y}(t)$, obtained by sampling all of the pixels at the same time. The main disadvantage of using such a fixed global sampling frequency is in the reduction of the dynamics of the changes detection of the continuous function of time $f_{x,y}(t)$.

This method, used in main stream current acquisition techniques, originates from the early times of audio signals. As shown in Fig.2.1(a) this process generates unnecessary redundant data when values are unchanged over a long period of time. This consuming process is acceptable if only few signals are to be considered, but requires an enormous amount of resources for a high number of simultaneous signals as it happens in the case of video streaming, where nowadays sensors can acquire

images up to millions of pixels. This sampling is incompatible with a compact representation of visual information, as it requires the acquisition, transmission, processing and storage of unnecessary redundant data.

In order to overcome these limitations and provide an accurate temporal sampling of $f(t)$, it is more efficient to detect variations of $f(t)$ just at the exact time at which they occur (Fig.2.1(b)), namely sampling on the amplitude axis.

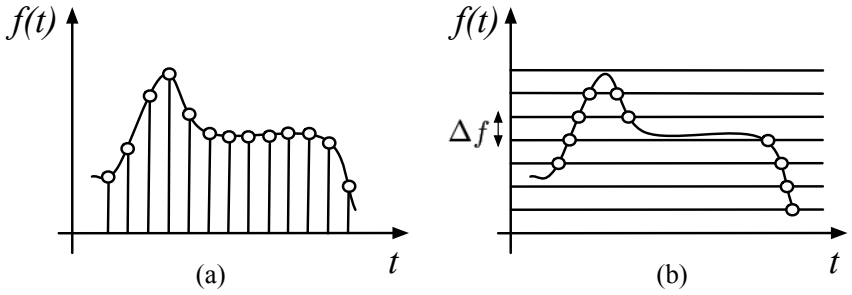


Figure 2.1: Two ways to sample functions values, in (a) using a classic constant scale on the t axis, in (b) using a constant scale but on the values of $f(t)$.

This process is data oriented and discards redundancies at the lowest level. Changes are detected precisely at the time they occur, overcoming all limitations of fixed time sampling. This signal oriented codification provides a compact representation of light changes and produces a sparse representation of data, as it is extremely rare that in natural environments the whole content of the visual field changes completely in the time between two consecutive frames.

In the worst case of cluttered and dynamic environment this data encoding will tend to the asymptotic value represented by the whole frame acquisition. **check with Ryad!!!**.

If $f(t)$ is quantized according to a predefined quantity Δf, it is possible to define a function Ev providing temporal events corresponding to the exact change time of $f(t)$ (Fig.2.1). The absolute value of luminance is no longer the element to be retrieved, the spatial and temporal locations of Δf changes are sufficient to give

an estimate of the visual data. Nevertheless, in the next section we will show that luminance can always be estimated using a incremental summation process.

This paradigm introduces a major road map change in computer vision, as it states that vision does not rely on a sampled set of low dynamics images but on a collection of asynchronous functions, each corresponding to a pixel which independently and asynchronously encodes changes of light at different spatial locations.

2.1.1.2 Codifications strategies to encode light changes

There are several ways to quantize $f_{x,y}$ according to its value. Let us define $\{t_k\}$, the set of times of the signal sampling. With the assumption that $\forall k \in \mathbb{N}, t_{k+1} > t_k$ and t_0 is the initial time, a standard set is:

$$T = \{t_k \mid |\mathcal{F}(f_{x,y}(t_k) - f_{x,y}(t_{k-1}))| = \Delta f\}\,. \tag{2.5}$$

The signal is sampled every time the variation of the magnitude of the function $\mathcal{F}$ is equal to Δf.

In principle there is no forward method to choose $\mathcal{F}$, that has to be selected according to the task to be performed. To study the general properties of codification based on relative changes, we will set $\mathcal{F}$ as the identity function. Unfortunately, there is no elegant formulation of T, as assumptions on f (to derive f^{-1}) contradict the random nature of light changes in scenes. Once T is set, $Ev(x, y, t)$ can be defined as:

$$Ev(x, y, t) = \delta(t, t_k)\dot{s}ign(f'_{x,y}(t)), \tag{2.6}$$

where $\delta()$ is the Kronecker delta function and $sign()$ is the sign function of a real number taking value in $\{-1, 1\}$.

$Ev(x, y, t)$ is a compact representation of $f_{x,y}$, its values are in the set $\{-1, 0, 1\}$:

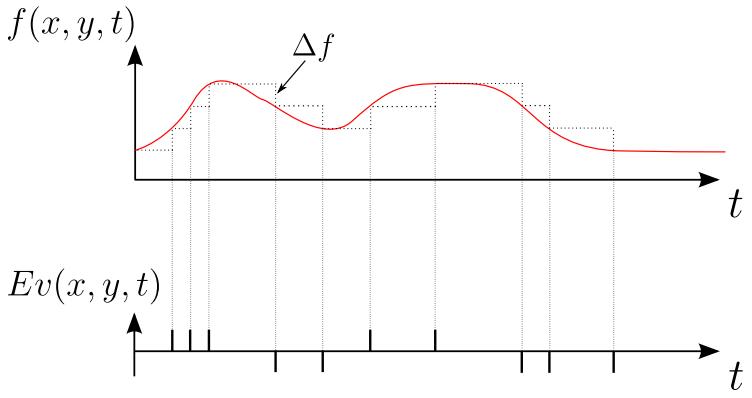


Figure 2.2: Codification of pixels gray-level variations into temporal contrast events.

$+1$ and -1 indicate a Δf change towards lighter or darker stimuli respectively, 0 corresponds to absence of change, or to changes smaller than Δf. Once Ev, Δf and $f_{x,y}(t_0)$ are known, $f_{x,y}$ can be approximated by a piecewise constant function $\widehat{f}_{x,y}$:

$$\widehat{f}_{x,y}(t) = \sum_{i=0}^{+\infty} f_i(t), \tag{2.7}$$

where

$$f_0(t) = \begin{cases} f_{x,y}(t_0) & \text{for } 0 \leq t < t_0 \\ 0 & \text{otherwise} \end{cases}, \tag{2.8}$$

$$f_i(t) = \begin{cases} Ev_{t_1,t_i}(x,y)\Delta f & \text{for } t_i \leq t < t_{i+1} \\ 0 & \text{otherwise} \end{cases}, \tag{2.9}$$

and

$$Ev_{t_m,t_n}(x,y) = \sum_{k=m}^{n} Ev(x,y,t_k). \tag{2.10}$$

As shown in Fig.2.2, $f'_{x,y}$ changes sign during the transitions, the next event appears only when the whole amount of change is larger than Δf from the last event.

$Ev(x,y,t)$ is a series of impulses, that allow a direct codification of the variation

of light of a scene in a more compact way than any frame acquisition process. This asynchronous architecture is one of the most general form of light acquisition. Frame images are clearly a special case, as shown in Eq. 2.7 the value of $\widehat{f}_{x,y}$ can be estimated any time needed. Frames can still appear spontaneously if the whole scene content changes completely at the same time. We will demonstrate farther that frames are most of the times not needed to perform visual tasks.

In what follows we omit the spatial variables x and y to define $Ev(t)$ as the set of $Ev(x, y, t)$ for $x \in [0, M-1], y \in [0, N-1]$:

$$Ev(t) = \{Ev(0, 0, t), \ldots, Ev(M-1, N-1, t)\} \tag{2.11}$$

and Ev_{t_1,t_2} is the spatiotemporal volume restricted on the interval $[t_1, t_2]$.

As described in the introduction, the Dynamic vision sensor is one of the first event-driven asynchronous sensors. It implements a specific case of discretization, where the function $f(t)$ is equal to the logarithm of light intensity. This specific function makes the pixel sensitive to local contrast change, rather than absolute light intensity, implementing a form of local gain control that adapts the gain of the pixel to the level of illumination and increases the sensor dynamic range (8). Unless explicitly mentioned, the work described in this thesis refers to the DVS and uses its recorded events for validating the developed computational methods. Figure 2.3 shows the values of $Ev(x, y, t)$ for all pixels responding to an horizontal translating bar. The output is part of a unique spatiotemporal volume, instead of being confined in separated frames.

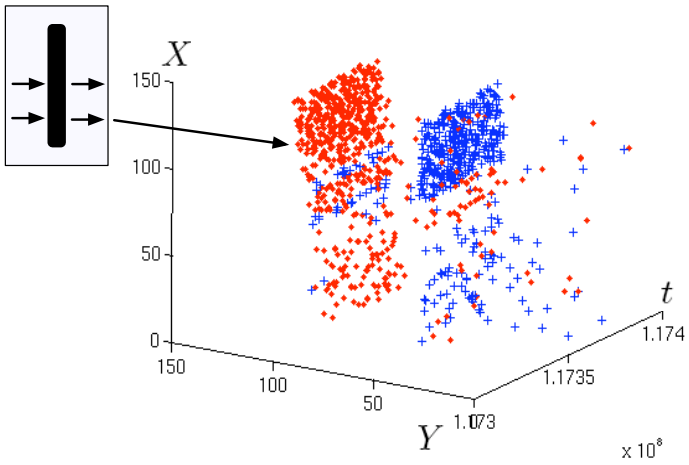


Figure 2.3: Codification of a horizontal translating bar using an asynchronous acquisition process: "·" and "+" correspond to the value -1 and $+1$, respectively, 0 is not represented.

2.1.2 Properties of event-driven acquisition

2.1.2.1 Decomposition of spatiotemporal volumes

The asynchronous acquisition process does not generate frames, but it is possible to generate images at any time if $Ev(t)$ is available. An image $\widehat{I}_{t_k,t_{k+1}}$ as a set of $\{\widehat{f}_{x,y}\}$ between $[t_k, t_{k+1}]$ can be generated according to Eq. 2.7. The content of $\widehat{I}$ is simply the sum of all $Ev(x, y, t)$ functions over the considered period of time. Images can then be generated when needed and at frequencies as high as the inverse of the pixels' elementary time step activation. The summation nature of Eq. 2.7 is a major property, as image processing and vision applications relying on such an acquisition process can rarely be limited by the bandwidth of the connection line or the computation power of computers. In networking applications, this means that if a common reference is set between two peers, the generation of images can follow the maximal speed rate of the communication lines.

Another interesting property of asynchronous image formation is shown in Fig.2.4. A spatiotemporal volume $Ev(t)$ restricted to the time interval $[t_0, t_{n+1}]$ can be written

as a sum of spatiotemporal sub-volumes :

$$Ev_{t_0,t_{n+1}}(t) = \sum_{k=0}^{n} Ev_{t_k,t_{k+1}}(t). \tag{2.12}$$

If images $\widehat{I}_{t_k,t_{k+1}}$ are built by summing events over the intervals $[t_k, t_{k+1}]$, we can write:

$$\widehat{I}_{t_0,t_{n+1}} = \sum_{k=0}^{n} \widehat{I}_{t_k,t_{k+1}}. \tag{2.13}$$

An image can be naturally decomposed as a summation of all generated previous images.

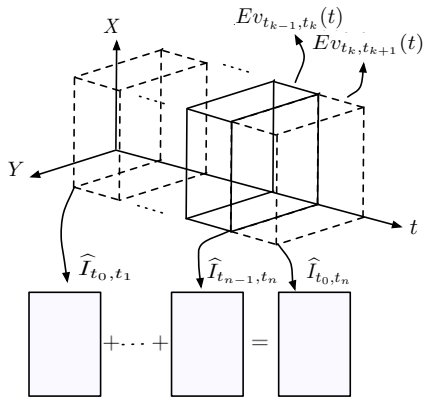


Figure 2.4: An asynchronous image generated at a time t_{n+1} can naturally be decomposed as a summation of all previously generated images.

2.1.2.2 Noise

Every pixel has a spontaneous impulse activity decorrelated from the content of the scene. Every pixel emits a noise impulse with a period of $15s$ and standard deviation of $1.5s$. The main source of spontaneous activity is a fixed pattern noise, mainly due to the leaky switch that resets the pixel after the generation of a spike. The variability of every pixel is highly unpredictable due to the mismatch in the silicon substrate. Fixed pattern noise can be removed by acquiring data from a static

scene for a long time interval (typically in the order of three minutes), providing a suppression of 92% of noise impulses. The remaining noise is due to other electronic noises, it can be easily removed by applying median filters.

2.1.2.3 Events and images

In order to provide a comparison with frame cameras, Fig.2.5 illustrates two sequences of images: a static acquisition of road traffic (Fig.2.5(a)) and a hand held moving camera moving on the sidewalk (Fig.2.5(b)); each image is created from event summation over a period of $50ms$.

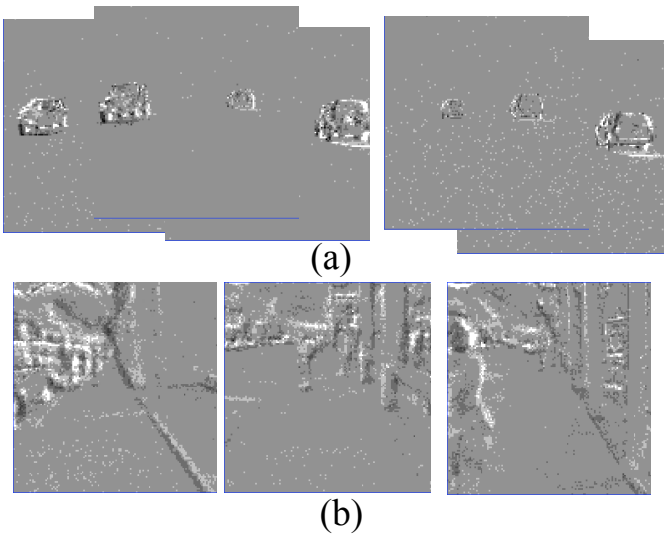


Figure 2.5: Generated images from event sequences for summation periods of 10ms (a) static camera observing road traffic, (b) a hand held moving camera on the sidewalk.

The static traffic acquisition sequence (a) is 1.75 minutes long, for a total of 876340 events, while the hand-held camera sequence (2) lasts 0.44 minutes, for a total of 1.153.600 events. Considering that 2 bits are needed to encode events polarity and the number of pixels is 128×128, the amount of acquired data is $213.9Kb$ for the first sequence and $281.6Kb$ for the second. The number of events depends on the content of the recorded scene. In the case of static acquisition only moving cars

and pedestrians can generate events, while in the case of hand-held camera also the relative motion of background produces events. For a comparison, we can calculate the amount of data produced by a traditional sensor with standard frame rate of 30 images/s: assuming that frames are standard 256 grey scale images of 128×128 pixels, the total amount of recorded data would be $49.12Mb$ for (a) and $12.36Mb$ for (b). The asynchronous sensor time resolution is $1\mu s$, which means that in principle an image sequence at a rate of $1\mu s$ can be generated from the collected data. To obtain the same temporal resolution with traditional digital cameras, a fast, 1GHz, camera should have been used. In this case the collected data would increase of more than 7 orders of magnitude. These results can be easily explained by the fact that acquiring images induces unnecessary data load, as most pixels do not change their content across consecutive frames. The hand-held camera produced a much larger amount of data than the static acquisition but it remains far from the data load produced if the same scene was acquired with a classic frame based camera, even at a much lower time resolution.

2.2 Event-based computation

The easiest way to perform visual tasks with event-based data is the application of standard computer vision algorithms on reconstructed frames from the stream of events. However, generating frames will only increase computational loads and lower real-time properties. The main interest of event driven acquisition relies on a wide temporal dynamic in the arrival of events, therefore it is essential to take into account this property and set up algorithmic processes that directly work on events. This is an important step as every event should require few computations to be able to be processed easily on the fly. To fully exploit the advantages of such framework, it is then important to understand the computational connection between current

events and the preceding ones to set up a fast integral computation.
In the following we introduce a framework for computing with events, starting from the definition of event-based convolution to the formalization of the computation of event-based optic flow.

2.2.1 Temporal convolution

We examine the event-based formulation of convolution, one of the most used standard signal processing operation, applied to spatiotemporal volumes of events. Let $*$ denotes the convolution operator applied to two functions g and $\hat{f}$ where $\hat{f}$ is the quantized form of an intensity function f introduced in Sec. 2.1.1.1. g is a function complying with the usually requested conditions for the standard convolution product. Finally, both functions are also assumed to be causal.
If $\hat{f}$ is acquired between t_0 and t_n, using results of Eqs. 2.7-2.10:

$$(g * \hat{f})(t,n) = (g * \sum_{i=0}^{n} f_i)(t) = \sum_{i=0}^{n} \int_{\mathbb{R}} g(t-u) f_i(u) du. \tag{2.14}$$

Sum and integration can be switched due to the finite summation from 0 to n. Finally, since each $f_i(t)$ is equal to 0 outside the interval $[t_i, t_{i+1}]$:

$$\begin{aligned}(g * \hat{f})(t,n) &= \sum_{i=0}^{n} \int_{t_i}^{t_{i+1}} g(t-u) f_i(u) du \\ &= \sum_{i=0}^{n} Ev_{t_1,t_i}(x,y) \Delta_f \int_{t_i}^{t_{i+1}} g(t-u) du\end{aligned} \tag{2.15}$$

Thanks to the linearity of the convolution operator, if an additional event is detected, the convolution product can be efficiently computed by adding the term

$(g * f_{n+1})(t)$ to the previous result. Convolution is calculated recursively:

$$\begin{aligned}(g * \hat{f})(t, n+1) =& (g * \hat{f})(t, n) \\ &+ Ev_{t_n, t_{n+1}}(x, y)\Delta_f \int_{t_i}^{t_{i+1}} g(t-u)du\end{aligned} \tag{2.16}$$

2.2.2 Spatial convolution

Spatial convolution follows the same principle described above, in this case g is a two-dimensional function and $\hat{f}$ is replaced by $\hat{I}_{t_0,t_n}$, the image built as explained in Eq. 2.13:

$$(g * \widehat{I}_{t_0,t_n})(x, y) = \int_{\mathbb{R}} \int_{\mathbb{R}} g(x-u, y-v)\hat{I}_{t_0,t_n}(u, v)dudv. \tag{2.17}$$

If an event appears at t_{n+1}, the convolution is updated recursively:

$$(g * \widehat{I}_{t_0,t_{n+1}})(x, y) = (g * \widehat{I}_{t_0,t_n})(x, y) + (g * \widehat{I}_{t_n,t_{n+1}})(x, y) \tag{2.18}$$

This is different from the slow and power-hungry computation required for frame-based convolution, event-based asynchronous convolution allows fast processing due to the light computational load that exploits the integral properties of the acquisition.

Chapter 3

Event based visual navigation

This chapter presents the approach developed in the context of the event-based paradigm to provide mobiles platforms the capability of autonomous navigation. We developed a navigation system based on the computation of the optical flow. The optical flow is the projection of motion in the observed scene onto the imager focal plane and conveys all the necessary information needed for reaching a point avoiding obstacles. In this chapter I describe the method developed for the computation of the optical flow using event driven sensors and the algorithms for calculating the focus of expansion, the detection of obstacles and the relative time to contact.

3.1 The optical flow

3.1.1 State of the art

Visual flow is a topic of several research fields that has been intensively studied since the early days of computational neuroscience. It is widely used in artificial vision and essential in navigation. Visual flow is known to be an ill-posed noisy visual measure, limited by the aperture problem. Its use in real-time applications on natural scenes is generally difficult and computationally expensive, it is usually

computed sparsely on highly salient points.

Visual flow techniques are commonly classified under one of four major categories:

- Energy-based or frequency-based methods estimate optical flow from the output of velocity-tuned filters designed in the Fourier domain ((44; 45; 46)),
- Phase-based methods estimate image velocity in terms of band-pass filter outputs ((47)),
- Correlation-based or region-matching methods search for a best match of small spatial neighborhoods between adjacent frames ((48; 49; 50; 51; 52; 53)),
- Gradient-based or differential methods use spatio-temporal image intensity derivatives and an assumption of brightness constancy ((54; 55; 56)),

Energy-based techniques's are slow (47) and not adequate for real-time applications where gradient-based approaches perform better, as they rely on correlations. Visual flow is generally slow and does not exceed several Hz on dense input. There are existing solutions to speed-up the computation according to a trade off between accuracy and efficiency (57). Preprocessing stages and kernel differentiation are often needed but they affect drastically real-time performance. In this case, accuracy is linked to the size of kernels that inevitably influences the execution time. If temporal kernels are used, then the buffering of images needed to perform computation dramatically increases the amount of stored data and introduces additional time delay in the computation.

The high computational cost of all of the approaches described above are not suitable for real-time applications. Frame-based flow computation using large temporal windows is not comparable with the temporal precision of biological sensors that respond with 1ms precision. The same observation applies to artificial vision

that intrinsically remains linked to the frequency of the available cameras, generally not exceeding 60Hz. Most of the developed techniques are computationally expensive and are mostly used off line.

Lee claims in (58) that the fundamental ecological stimulus for vision is not a camera like time static images but a constantly changing optic array or flow field, the description of which must be in spatio-temporal terms.

In the following, I present my research on the event-based computation of the optical flow. Two different methods have been developed and tested. The former method introduces a new formulation entirely and only based on events' timing. The optical flow is obtained by adapting the differential brightness consistency constraint to event-based framework. Local image intensities are then approximated by events' summations, as the DVS does not provide absolute intensities. This method has a major weakness in the noise and drift in the computation of absolute illumination levels, needed for the computation of spatial derivatives. This is overcome by an alternative method that offers a pure event-based time oriented computation of the motion flow within the focal plane that does not need to reconstruct intensity levels of illumination.

3.1.2 Asynchronous frameless event-based flow

3.1.2.1 Assumption and equations of the optical flow

The main assumption for the evaluation of the optical flow is the invariance of light intensity captured by the retina undergoing a small motion over an infinitesimally short duration. If we assume that the brightness of a small surface patch is not changed by motion, then expansion of the total derivative of brightness leads to

(59):

$$\begin{aligned}\frac{dI(x,y,t)}{dt} &= \frac{\partial I}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial I}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial I}{\partial t} \\ &= \begin{pmatrix} \frac{\partial I}{\partial x} & \frac{\partial I}{\partial y} \end{pmatrix}^t \begin{pmatrix} \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial t} \end{pmatrix} + \frac{\partial I}{\partial t} \\ &= grad^t(I) \begin{pmatrix} v_x \\ v_y \end{pmatrix} + \frac{\partial I}{\partial t} = 0,\end{aligned} \tag{3.1}$$

where is I is the image containing gray levels. This equation is solved for the velocity vector $(v_x, v_y)^t$ but it is under-determined, since two variables must be estimated, given only one equation. One of the most popular techniques (60) to overcome this problem is based on the assumption of local constant flow: $(v_x, v_y)^t$ is constant over the neighborhood of pixel $(x, y)^t$. If the neighborhood is a $n \times n$ window, $m = n^2$ optical flow equations can be written:

$$\begin{pmatrix} grad^t(I(x_1, y_1)) \\ \vdots \\ grad^t(I(x_m, y_m)) \end{pmatrix} \begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} -I_{t_1} \\ \vdots \\ -I_{t_m} \end{pmatrix}, \tag{3.2}$$

with $gradt(I)$ being the spatial gradient and I_t the partial temporal derivative of I. The equation system can then be solved using least square error minimization techniques.

This constraint can be formulated in an event-based manner. The main difficulty being that the used sensor does not provide gray levels needed for the computation of spatial derivatives, obtained by comparing illumination levels of neighboring pixels. Nevertheless, for adjacent active pixels it is possible to provide an estimation of the

spatial gradient by comparing their instantaneous activities:

$$\begin{cases} \dfrac{\partial e(x,y,t)}{\partial x} \sim \displaystyle\sum_{t-\Delta t}^{t} e(x,y,t) - \sum_{t-\Delta t}^{t} e(x-1,y,t) \\ \dfrac{\partial e(x,y,t)}{\partial y} \sim \displaystyle\sum_{t-\Delta t}^{t} e(x,y,t) - \sum_{t-\Delta t}^{t} e(x,y-1,t) \end{cases}. \tag{3.3}$$

Δt is a temporal interval of few μs, generally set to $50\mu s$. The estimation of the temporal gradient is expected to be more precise than in the frame-based framework due to the temporal precision of the DVS. It can be written as:

$$\frac{\partial e(x,y,t)}{\partial t} \sim \frac{\sum_{t-\Delta t}^{t1} e(x,y,t) - \sum_{t-\Delta t}^{t} e(x,y,t)}{t - t_1}, \quad \text{with } t_1 < t \tag{3.4}$$

By substituting Eq. 3.3 and Eq. 3.4 into Eq. 3.2, the j^{th} line of the matrix equality can be reformulated using events:

$$\begin{gathered} \left(\sum_{t-\Delta t}^{t} e(x_j,y_j,t) - \sum_{t-\Delta t}^{t} e(x_j-1,y_j,t)\right) v_x \\ + \left(\sum_{t-\Delta t}^{t} e(x_j,y_j,t) - \sum_{t-\Delta t}^{t} e(x_j,y_j-1,t)\right) v_y \\ = \frac{\sum_{t_1}^{t} e(x_j,y_j,t)}{t-t_1} \end{gathered} \tag{3.5}$$

since

$$\sum_{t-\Delta t}^{t1} e(x,y,t) - \sum_{t-\Delta t}^{t} e(x,y,t) = \sum_{t_1}^{t} e(x,y,t).$$

The general optic flow algorithm is detailed by the **Algorithm 1**.

Algorithm 1 Event-based optical flow

For each incoming e(x,y,t):
Define a $(n \times n \times \Delta t)$ neighborhood around $(x, y, t)^T$
Compute the partial derivatives:

- $\frac{\partial e(x,y,t)}{\partial x}$, $\frac{\partial e(x,y,t)}{\partial y}$, $\frac{\partial e(x,y,t)}{\partial t}$

Solve Eq. (3.2) written using Eq. (3.5) for $(v_x, v_y)^t$

3.1.3 Event-based visual flow

3.1.3.1 Flow definition

The stream of events from the DVS can be mathematically defined as follows: let $e(\mathbf{p}, t) = (\mathbf{p}, t)^T$ a triplet giving the position $\mathbf{p} = (x, y)^T$ and the time t of an event. We can then define the function Σ_e that maps to each $\mathbf{p}$, the time t:

$$\begin{array}{rcl} \Sigma_e : \mathbb{N}^2 & \rightarrow & \mathbb{R} \\ \mathbf{p} & \mapsto & \Sigma_e(\mathbf{p}) = t. \end{array} \tag{3.6}$$

Time being an increasing function, Σ_e is then a monotonically increasing surface.

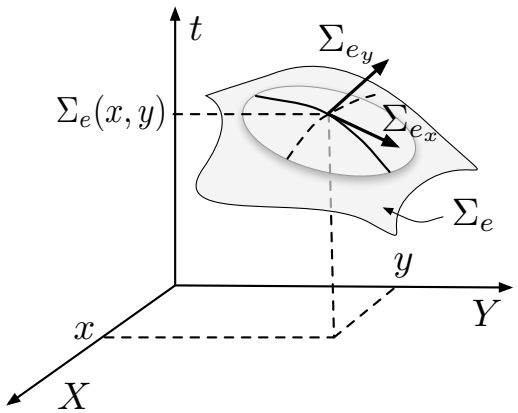


Figure 3.1: General principle of visual flow computation, the surface of active events Σ_e is derived to provide an estimation of orientation and amplitude of motion.

We then set the first partial derivatives with respect to the parameters as: $\Sigma_{e_x} =$

$\frac{\partial \Sigma_e}{\partial x}$ and $\Sigma_{e_y} = \frac{\partial \Sigma_e}{\partial y}$. We can then write Σ_e as:

$$\Sigma_e(\mathbf{p} + \Delta\mathbf{p}) = \Sigma_e(\mathbf{p}) + \nabla\Sigma_e^T \Delta\mathbf{p} + o(||\Delta\mathbf{p}||), \tag{3.7}$$

with $\nabla\Sigma_e = (\frac{\partial \Sigma_e}{\partial x}, \frac{\partial \Sigma_e}{\partial y})^T$.

The partial functions of Σ_e are functions of a single variable whether x or y. Time being a strictly increasing function, Σ_e is a nonzero derivatives surface at any point. It is then possible to use the inverse function theorem to write around a location $\mathbf{p} = (x, y)^T$:

$$\begin{aligned} \frac{\partial \Sigma_e}{\partial x}(x, y_0) &= \frac{\mathrm{d}\Sigma_e|_{y=y_0}}{\mathrm{d}x}(x) = \frac{1}{v_x(x, y_0)}, \\ \frac{\partial \Sigma_e}{\partial y}(x_0, y) &= \frac{\mathrm{d}\Sigma_e|_{x=x_0}}{\mathrm{d}y}(y) = \frac{1}{v_y(x_0, y)}, \end{aligned} \tag{3.8}$$

$\Sigma_e|_{x=x_0}$, $\Sigma_e|_{y=y_0}$ being Σ_e restricted respectively to x_0 and y_0. The gradient $\nabla\Sigma_e$ can then be written:

$$\nabla\Sigma_e = (\frac{1}{v_x}, \frac{1}{v_y})^T. \tag{3.9}$$

The vector $\nabla\Sigma_e$ measures the rate and the direction of change of time with respect to the space, its components are also the inverse of the components of the velocity vector estimated at $\mathbf{p}$.

3.1.3.2 Flow regularization

The flow definition given by Eq. 3.9 is sensitive to noise since it consists in estimating the partial derivatives of Σ_e at each individual event. One way to make the flow estimation robust against the noise is to add a regularization process to the estimation. To achieve this, we assume a local velocity constancy. This hypothesis is satisfied in practice for small clusters of events. It is then equivalent to assume Σ_e being locally planar since its partial spatial derivatives are the inverse of the speed,

hence constant velocities produce constant spatial rate of change in Σ_e. Finally, the slope of the fitted plane with respect to time axis is directly proportional to the motion velocity. The regularization also compensates for absent events in the neighborhood of active events where motion is being computed. The plane fitting provides an approximation of the timing of still non active spatial locations due to the imperfection of the sensor.

A robust plane fitting is applied to each event arriving at time t over a spatio-temporal window of dimensions $L \times L \times 2\Delta t$, centered on the event. In practice, a spatio-temporal window is selected empirically for the fitting, in what follows $L = 3$ and $\Delta t \sim 1ms$. This setting produces accurate result if compared to the ground-truth.
Any event $e(\mathbf{p}, t)$ belongs to a plane of parameters $\mathbf{\Pi} = \begin{pmatrix} a & b & c & d \end{pmatrix}^T$ if the following equality is satisfied:

$$\mathbf{\Pi}^T \begin{pmatrix} \mathbf{p} \\ t \\ 1 \end{pmatrix} = 0. \tag{3.10}$$

According to this equality, the regularization operation can be performed as detailed in Algorithm 2 that provides the whole approach of computing motion flow:

The threshold in step 4 can be set to th_1 = 1e-5, it is usually the magnitude of accuracy we get from this iterative estimation algorithm. The second threshold in step 5 is also set to $th_2 = 0.05$, according to experimental results. Moreover, the error usually converges in only 2 to 3 iterations, in short the resulting algorithm is robust and consumes little computational resources, as shown in sec4.1 where

Algorithm 2 Local planes fitting algorithm on incoming events

1: **for all** event $e(\mathbf{p}, t)$ **do**
2: Define a spatio-temporal neighborhood Ω_e, centered on e of spatial dimensions $L \times L$ and duration $[t - \Delta t, t + \Delta t]$.
3: Initialization:

- apply a least square minimization to estimate the plane $\mathbf{\Pi} = \begin{pmatrix} a & b & c & d \end{pmatrix}^T$ fitting all events $\tilde{e}_i(\mathbf{p}_i, t_i) \in \Omega_e$:

$$\tilde{\mathbf{\Pi}}_0 = \underset{\mathbf{\Pi} \in \mathbb{R}^4}{\operatorname{argmin}} \sum_i \left| \mathbf{\Pi}^T \begin{pmatrix} \mathbf{p}_i \\ t_i \\ 1 \end{pmatrix} \right|^2 \tag{3.11}$$

- set ϵ to some arbitrarily high value ($\sim 10e6$).

4: **while** $\epsilon > th_1$ **do**
5: Reject the $\tilde{e}_i \in \Omega_e$ if $|\tilde{\mathbf{\Pi}}_0^T \begin{pmatrix} \mathbf{p}_i \\ t_i \\ 1 \end{pmatrix}| > th_2$ (i.e. the event is too far from the plane) and apply Eq. 3.11 to estimate $\tilde{\mathbf{\Pi}}$ with the non rejected $\tilde{e}_i$ in Ω_e.
6: Set $\epsilon = ||\tilde{\mathbf{\Pi}} - \tilde{\mathbf{\Pi}}_0||$, then set $\tilde{\mathbf{\Pi}}_0 = \tilde{\mathbf{\Pi}}$.
7: **end while**
8: Attribute to e the velocity defined by the fitted plane.
9: **end for**
10: **return** $v_x(e)$, $v_y(e)$.

both optical flow algorithms are validated and tested against standard frame-based camera.

3.2 Focus of expansion

3.2.1 Definition

The focus of expansion is the single (focal) point, in space, where all divergent optical flow vectors originated by translation of the sensor intersect. According to Gibson's Ecological Theory of Perception(43), a more generic designation of the focus of expansion would be *Dynamic Ambient Optical Array* which represents the single point in space where the optical information converges. The focus of expansion plays an important role in many vision applications such as three-dimensional reconstruction, range estimation, time to contact and obstacle avoidance. In our context this point is looked for motion recovery, specifically of the translational direction of the robot on which the vision system is embedded.

3.2.2 State of the art

State of art methods for the computation of the focus of expansion can be divided in four main categories: direct, discrete, differential and continuous.

- *Direct approach*: This approach aims to determine the motion of the environment directly from the variation of the image brightness pattern in a motion sequence. It gave rise to different ideas, i.e, the computation of the FOE using unequally constraint applied to features in two frames(61), the direct recovery of the FOE by imposing the geometric constraint that depth is positive(62), or the computation based on the information from various local regions in the image, each voting for a direction toward the FOE(63).

- *Discrete approach*: The information about the movement of brightness patterns at only a few points is used to determine the motion: it matches discrete points in a sequence of images. This technique needs to track or identify features in a set of images taken at different times(64).

- *Differential*: This method uses the first and second spatial partial derivatives of the optic flow. (65) claims this is sufficient to know those information at a single point to uniquely determine the motion.

- *Continuous*: In this method the whole optical flow is used. It takes advantage of the abundance of available data to compute a robust numerical results(66)(67)

Applying traditional approach, which requires the estimation of the induced image motion and calculation of the camera motion from it, i.e. The optical flow, is generally said to be inefficient and computationally expensive. In this section I show that this statement only concerns the frame-based methods.

3.2.3 Equations

In our method, every flow vector provides an estimation of the location of the FOE in the visual field. We consider a probability map of the visual field, where each flow event votes for a probable position of the FOE in the plane behind the vector, as shown in Fig3.2a-b. In this map, each point represents the likelihood of the FOE to be located at the corresponding location in that field. This method uses the redundancy of the flow pattern to smooth the noisy FOE estimation and reduce errors. In brief, our algorithm for estimating FOE is as follow: 1. Update the map with every computed flow, 2. Find the patch of visual field map with maximum value, 3. Shift the FOE toward the maximum patch. Our results show that the estimation of the FOE is accurate if we receive a sufficient number of events from

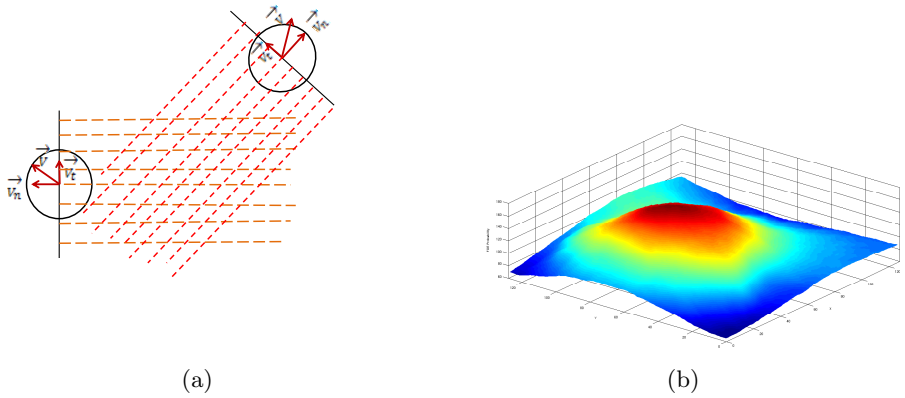
the sensor.

For an incoming event $e(\mathbf{p_e}, t_e)$ with a velocity vector v_e, we can define the set of spatial locations such as:

Algorithm 3 Computation of the Focus of expansion

Require: $M_{activity} \in \mathbb{R}^m \times \mathbb{R}^n$ and $M_{time} \in \mathbb{R}^m \times \mathbb{R}^n$. Set all values of M_{space} and M_{time} to 0

1: **for** every incoming $e(\mathbf{p_i}, t_i)$ at velocity $\mathbf{v_i}$ **do**
2: Determine all spatial location $\mathbf{p}$ such as $(\mathbf{p} - \mathbf{p_i}).\mathbf{v_i} < 0$
3: for all $\mathbf{p}$: $M_{activity}(\mathbf{p}) = M_{activity}(\mathbf{p}) + 1$ and $M_{time}(\mathbf{p}) = t_i$
4: $\forall\ \mathbf{p} \in \mathbb{R}^m \times \mathbb{R}^n$, update the value of $M_{activity}(\mathbf{p}) = M_{activity}(\mathbf{p}) e^{-\frac{t_i - M_{time}(\mathbf{p})}{\tau}}$
5: Find $\mathbf{p}_{FOE}$ the spatial location of the maximum value of $M_{activity}$ corresponding to the FOE location
6: **end for**



(a) (b)

Figure 3.2: Computation of the focus of expansion (a) the focus of expansion lies under the normal flow, we can then vote for a area of the focal plane shown in (b) the FOE is the max of this area

3.3 Time to contact

3.3.1 Definition

Time-to-Contact (TTC) is the time needed to reach a point under unknown constant relative velocity. Part of the attraction of TTC is that the calculation relies only

on image measurements and does not require camera calibration, knowledge of the structure of the environment nor the size of the obstacles. It is commonly used for obstacle avoidance of autonomous vehicles in dynamic environments. With TTC the observer is able to judge when to alter course without making any further estimates. TTC is a biologically inspired method for obstacle detection and reactive control of motion that does not require scene reconstruction or 3D depth estimation(68). Biological evidence shows that the TTC is in form of relative distance. For visual navigation this implies that the temporal units do not require camera calibration.

3.3.2 State of the art

TTC can be defined as the ratio of the distance z of an object and the relative speed of approach between an agent and an obstacle(69). This definition naturally suggests the use of the optical flow for the computation of the TTC(36)(70). However, as the computation of the optical flow under the frame-based approach is computationally expensive, other methods have been developed, for example using active contour affine scale(68). Where TTC can be estimated as the distance between two image points divided by the rate of change in that distance. In (71) a method using only accumulated sums of suitable products of image brightness derivatives is proposed. Claiming that the approach does not need features, object detection nor optical flow estimation, the author points out that the methods for estimating optical flow are iterative and need to work at multiple scales, which tends to be computationally expensive and require a significant effort to implement properly. (72) proposes that the TTC can be directly measured from a spatio-temporal image sequences obtained from uncalibrated camera. The method is based on the idea of measuring the rate of change of the "intrinsic scale", rapidly determined using a multi-resolution pyramid, and on the observation that the local size of features in an image may be directly measured from the scale of extremal points in a Laplacian scale space.

Another method (73) makes use of temporal derivative of the area of a closed active contour and proposes to avoid the problems associated with the computation of image velocity fields and their derivatives.

In the following, I present a method of computation of the time to contact relying on event-based computation of the optical flow which I show to be accurate and computationally efficient in sec4.1. This method relates to biological evidence, where the computation of the TTC in human looming relies on the estimation of the optical flow and its first derivative(74)(70)(75)

3.3.3 Equations

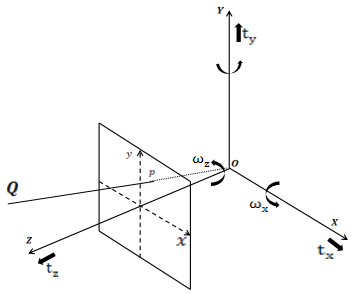


Figure 3.3: General principle of time to contact computation

In this section, we show how to estimate time to contact to an object given the FOE and optical flow. Consider a camera-centered coordinate system where the z-axis aligns with the line of sight. As shown in Fig.3.3, if the camera moves with translational velocity $\mathbf{t} = (t_x, t_y, t_z)^T$ and rotational velocity $\omega = (\omega_x, \omega_y, \omega_z)^T$ around its origin, then, the 3D velocity of a world point, $\mathbf{P} = (X, Y, Z)^T$, is :

$$\dot{\mathbf{P}} = -\mathbf{t} - \omega \times \mathbf{P} \tag{3.12}$$

or, in components,

$$\dot{\mathbf{P}} = \begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = - \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} - \begin{pmatrix} \omega_y Z - \omega_z Y \\ \omega_z X - \omega_x Z \\ \omega_x Y - \omega_y X \end{pmatrix}. \tag{3.13}$$

Assume the point $\mathbf{P}$ is projected onto the point $\mathbf{p} = (x, y)^T$ in the image plane. Applying the perspective projection, its image coordinates will be $x = \frac{X}{Z}$ and $y = \frac{Y}{Z}$. Temporal differentiation of image coordinates results is 2-D motion or velocity field induced in the image plane:

$$\dot{x} = \frac{(\dot{X}Z - X\dot{Z})}{Z^2}, \ \dot{y} = \frac{(\dot{Y}Z - Y\dot{Z})}{Z^2}. \tag{3.14}$$

By substituting $\dot{X}$ and $\dot{Y}$ from (3.13), the 2-D motion field can be written as:

$$\dot{\mathbf{p}} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \frac{1}{Z} \begin{pmatrix} xt_z - t_x \\ yt_z - t_y \end{pmatrix} + \begin{pmatrix} xy & -(x^2+1) & y \\ y^2+1 & -xy & -x \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \tag{3.15}$$

Then from 0, for a pure translational motion, i.e. $\omega = (0, 0, 0)^T$, the velocity field at each point will be:

$$\dot{\mathbf{p}} = \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \frac{t_z}{Z} \begin{pmatrix} x - \frac{t_x}{t_z} \\ y - \frac{t_y}{t_z} \end{pmatrix} \tag{3.16}$$

$(\frac{t_x}{t_z}, \frac{t_y}{t_z})$ actually represents FOE. In addition, the variable $\tau = \frac{Z}{t_z}$ is the time it takes for an object moving at constant velocity t_z to cross the distance Z, this is known as TTC. Thus, from (4), TTC can be computed by: $\tau = \frac{d_{foe}}{|\dot{\mathbf{p}}|}$ where d_{foe} is the distance of the considered point on the image plane and $|\dot{\mathbf{p}}|$ is the magnitude of computed optical flow at the same point. To compute TTC, we use the above

equation and apply it for every pixel. To make the algorithm less sensitive to the noise, we incrementally smooth the results using an online mean with the close neighborhood.

Chapter 4

Results and validation

In the following I present the performance of the algorithms introduced in chapter 3. First I validate the two event-based optical flow methods described above, comparing their performance with respect to a frame-based method, by assessing the precision of amplitude and direction of the calculated optical flow vectors, its temporal precision and, finally, its computational cost. Finally, the computation of the focus of expansion and time to contact will be tested. In the following experiments, the event-driven input is provided by a DVS camera connected to a PC via a USB port. The DVS has a temporal resolution of 1us. The motion flow algorithms have been implemented in C using a Linux driver of the DVS.

4.1 The optical flow

To assess the precision of amplitude and direction of the calculated optical flow vectors, the methods use the same two experiments, each of them involving a different setup. In a first experiment, which aims to validate the orientations, the motion flow is computed for a black bar painted on a white disk, rotating with a constant angular velocity, as shown in Fig.4.1. In a second experiment, the amplitudes of the

estimated optical flow are computed for a moving pattern of bars presented on a moving conveyor belt whose translational speed can be accurately set by adjusting the supply voltage of a DC motor (Fig.4.2).

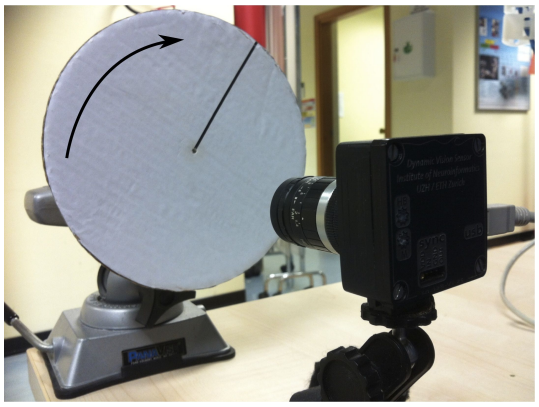


Figure 4.1: Experimental setup consisting of a black bar printed on a white disk, driven by a DC-motor.

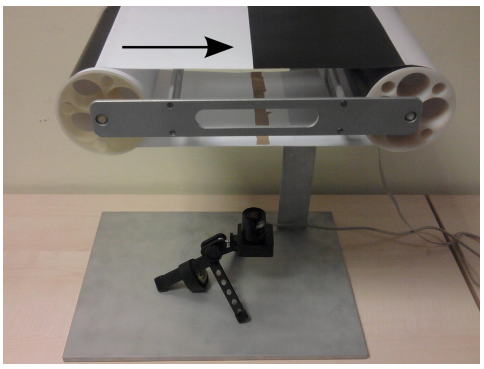


Figure 4.2: Belt driven by a DC-motor, controlled in closed-loop

4.1.1 Method 1: Asynchronous frameless event-based optical flow

The optical flow is computed using a neighborhood of 5×5 pixels for a $\Delta t = 50\mu s$.

Orientation To assess the computed orientations, a constant angular velocity of $\omega \sim 13,6 rad/s$ is set to the setup presented in figure 4.1. The events-based optical

flow is computed in real-time using the continuous train of output events. The frame-based optical flow is computed for a 60 fps emulated camera, obtained by accumulating events for $16.7ms$. Fig.4.4 shows the orientation of the bar computed from the flow obtained by the two techniques and is plotted versus the ground truth orientation of the bar. The event-based computation takes full advantage of the high temporal accuracy of the sensor, by providing a dense and accurate estimation of the rotation angles in real time.

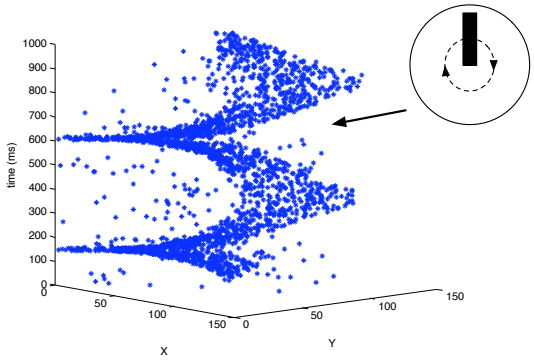


Figure 4.3: Space-time representation of events generated in response to a rotating black bar. Each dot represents a DVS event.

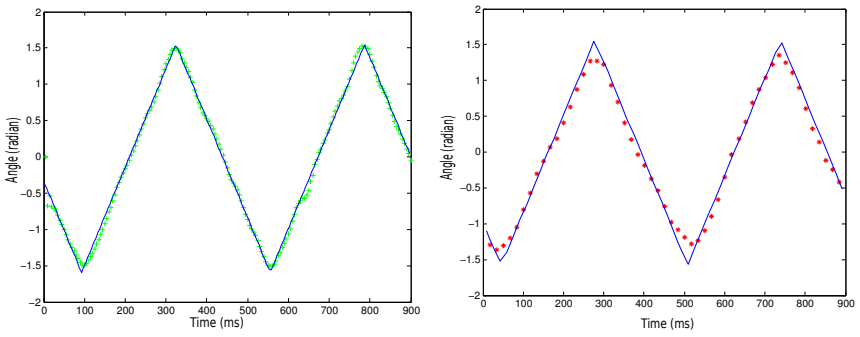


Figure 4.4: Bar orientation estimated by the events-based (+) and by the frame-based (∗) optical flow algorithms versus the true bar orientation over the time (−) for the input shown in Fig.4.3.

Amplitude In this experiment the magnitudes of the optical flow are computed for different known belt speeds obtained by driving the DC motor, figure 4.2, with supply voltages ranging from 500 to 1500 mV corresponding to a range of velocities of 30,83 – 83,05cm/s. Fig.4.5 shows the mean estimated amplitudes, as expected, there is a linear relationship between the ground truth and the computed velocity.

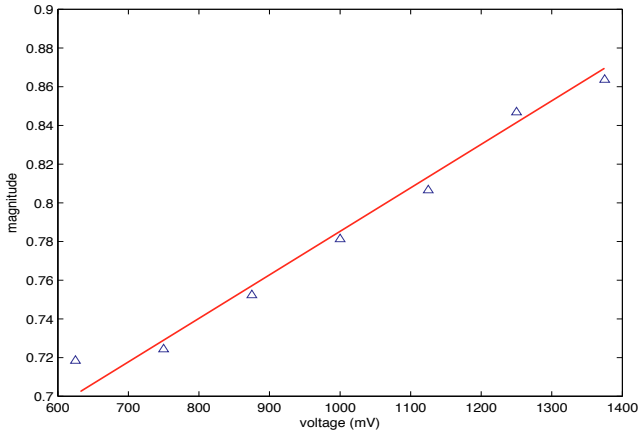


Figure 4.5: Estimated amplitudes of the optical flow versus the DC voltage ranging from 500 to 1500 mV apply on the setup presented in figure 4.2 and corresponding to a range of velocities of 30,83–83,05cm/s.

Temporal precision With this experiment I highlight the high temporal precision of the estimated optical flow. The stimulus consists of a grid of LEDs that turn on and off sequentially with a precisely tunable frequency. The distance between LEDs is 3.4cm and the time interval between the switching of successive LEDs is adjustable from 1ms to 1s and is set to 16ms in the current experiment.

The aim of the experiment is to estimate the rate of switching on of successive LEDs on the panel (or the apparent speed of the LED sequence). The experiment consists in estimating the temporal interval of the switching between two consecutive LEDs, the apparent speed is then the temporal interval divided by the distance between the LEDs. The switching on of a LED is detected by sensing the characteristic divergent pattern of the optical flow that it creates, as shown in Fig.4.6(b).

This divergent characteristic pattern of optical flow is created because, as the LED turns on, the pixel that views the brightest point in the LED reaches a threshold of activity first, followed later by the surrounding pixels.

In the experiment the DVS observes the panel continuously and for every incoming event the optical flow is computed. In a second stage, in order to detect precisely the exact timing of the LED's activation, a spatial vector pattern similar to Fig.4.6(b) is used as a template and correlated with the measured optic flow pattern in the neighborhood of the active pixel's location. In the experiment, the vector flow pattern of activation is set to a size of 5×5 vectors.

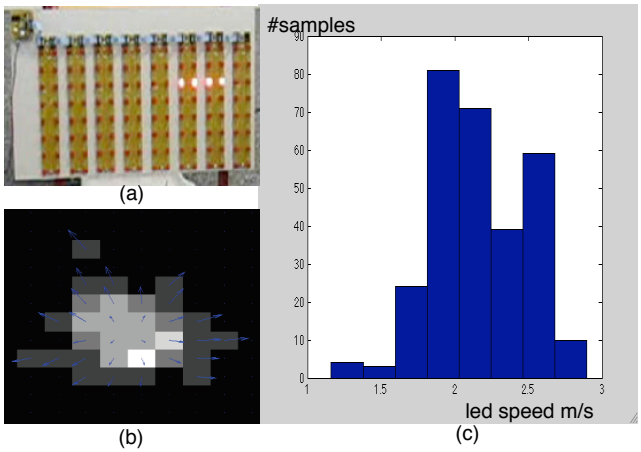


Figure 4.6: Computation of the optical flow generated by sequential activation of LEDs on a panel. Long exposure of this image shows 4 LEDS that have been turned on successively from left to right(a). (b) Switching on of a single LED generates a characteristic, expansion pattern of optic flow. (c) Distribution of estimated speeds of moving display created by sequential switching of LEDs in m/s, giving a mean value of $2.16m/s$, compared to an actual speed of $2.12m/s$.

Once the location of the active LED is determined, its position on the panel is retrieved and the estimation of the speed of activation of pixels can then be performed. The precise moment of switching on of an LED is established by detecting the divergent pattern of optic flow that will initially cover a small area and then grow continuously. The instant of switching on is then defined as the time when the central pixel is first detected and has reached threshold. Fig.4.6(c) shows the dis-

tribution of estimated speeds for a $2s$ sequence, giving an estimated mean speed of $2.169m/s$ while the ground truth speed is $2.125m/s$. The observed estimation errors are due to small misalignments of individual LEDs in the panel, small geometrical imperfections in the sensor and distortions in the optics of the camera.

Computational cost Tests were performed to compare the computational times required by the frame-based and the event-based methods for computing optic flow. These results were obtained using a C++ implementation of the algorithm, on an Intel Core 2 Duo 2.40GHz processor.
The optical flow was computed for a rotating bar as shown in Fig.4.3. The event-based method uses raw events, while the frame-based method uses frames acquired at a rate of $60Hz$ created by summing events. The following computation times were obtained:

- frame-based, $60fps$ ($16.7ms$): the number of processed pixels at each step is 16384, and a mean computation time is $251.7ms$,
- event-based (1e-$3ms$) : computation time for a period of $16.7ms$ corresponding to a mean number of events of 1340, the computation time is $9.65ms$, while the mean computation time of an event is 7.2e-$3ms$.

It takes $251.7ms$ to compute the optical flow between two successive frames on a time interval between successive frames of $16.7ms$. This means that in absence of any optimization, the optical flow can computed only at a rate of $4fps$. The optical flow computation is taking as long as 251.7 milliseconds because we are computing the flow at all of the 16384 pixel locations in the imager without any optimization in both frame and event-based. The event-based optical flow requires $7.2\mu s$ to compute the optical flow for each event, thus leading to a total time of $1340\times$(7.2e-3) =

$9.65ms$. Fig.4.7 shows that the event-based method delivers flow measurements at a cost that is approximately 25 times lower than does the frame-based method.

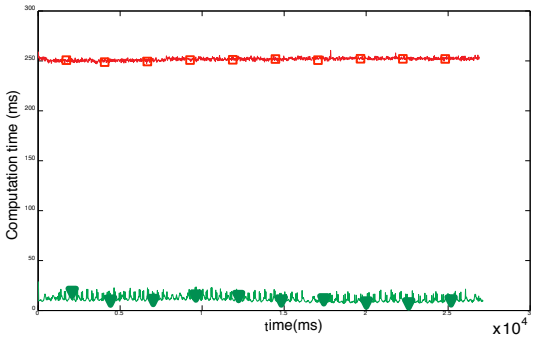


Figure 4.7: Computation time of event-based (triangles) vs frame based (squares) for the same observed sequence. The event-based method delivers flow measurements at a cost that is approximately 25 times lower than does the frame-based method.

Samples of computed flows This section presents examples of optic flow computed using the events-based method developed in the section 3.1.2. Fig.4.8 presents the decomposition of fast arms movements and their corresponding estimated optical flows. Fig.4.9 shows the optic flow associated with a bouncing ball.

4.1.2 Method 2: Event based motion flow

The results obtained with the event-based method are compared with frame-based algorithms such as Horn and Shunck (55), and Lucas and Kanade (76). For Horn and Schunk, following (47) the smoothness term λ is set to 0.5, with a number of iteration below 100. The Event-based flow computation is performed according to Algorithm 2. The flow is regularized as detailed, but Σ_e itself is never updated by the regularization operation. To avoid overlaping issues of the plan fitting on ON and OFF events because of thin textures, the events of the DVS are processed separately given their polarity. Results shown are merging both pathways.

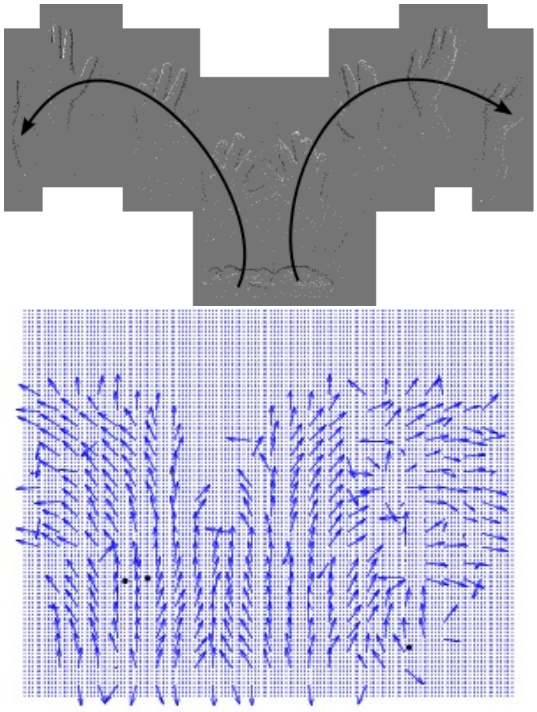


Figure 4.8: Optic flow for curved motion: (up) decomposition of the movement of two hands moving from the bottom to the sides of the scene and its (down) corresponding cumulative estimated optic flow over a time period of $1.5s$ on all the focal plane.

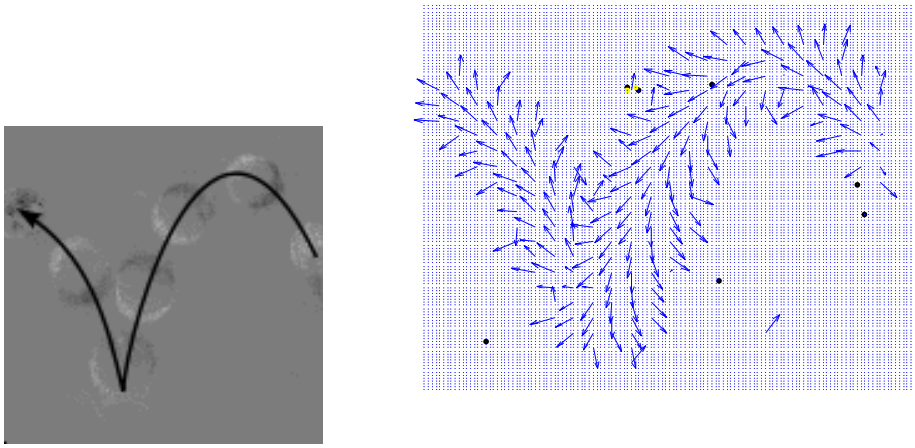


Figure 4.9: Optic flow for curved motion: (up) decomposition of the movement of a bouncing ball, (down) corresponding cumulative estimated optic flow over a time period of $0.8s$ on all the focal plane.

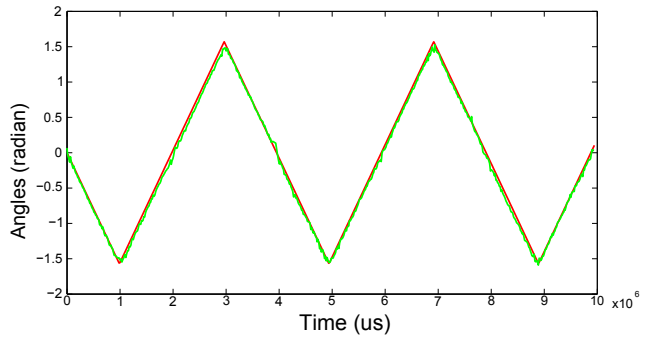
For indoor experiments, the illumination condition is the standard office lighting (i.e. 300 to 500 lux) from a non flickering source (halogen lamp). For outdoor experiments, the sequences are captured under the sunlight of an overcast day ($\sim$ 1000 lux).

Orientation To assess the precision of the flow orientation estimation, we use the setup shown in figure 4.1 with a constant angular velocity $\omega = 1.59\text{rad} \cdot \text{s}^{-1}$. The event-based optical flow is computed in real-time directly using the output train of events.

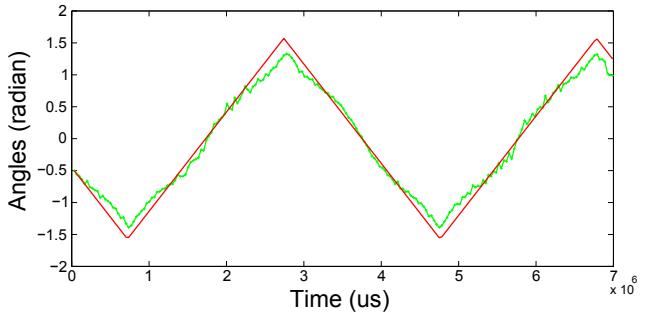
The frame-based optical flow is computed for a 30 fps using a conventional frame-based camera where images are rescaled to 128×128 pixels to allow for comparison with the DVS.

Fig.4.10 shows the orientation of the bar computed from the flow plotted versus the ground truth orientation of the bar for the event-based and Horn and Schunk algorithms. The mean error corresponding to the event-based algorithm is 0.037 rad, with a standard deviation of 0.025. The mean error corresponding to the Horn and Schunk (55) frame-based implementation is 0.113 rad, with a standard deviation of 0.078, this is 3 times higher than the event-based method. The event-based computation takes full advantage of the high temporal accuracy of the sensor, by providing a smoother and more accurate estimation of the rotation angles in real time. A sample of the computed flow is shown in Fig.4.11, highlighting the quality of the estimated motion and it is shown for a period of time of 3 ms. The lower accuracy of the flow estimated by the Horn and Shunk algorithm on reconstructed images is largely due to the lower temporal resolution of frames. Increasing the frame-rate would of course improve the frame-based performance, but at the cost of an increase of the needed computational resources and, as shown later, at the cost

of real-time performances.



(a)



(b)

Figure 4.10: (a) Ground truth and computed orientation for the stimulus in Fig.4.1. In red the real orientation of the bar, in green the estimated one for the event-based method. (b) Angles estimated using frame-based Horn and Schunk (55). The real angles are plotted in red, while the estimated ones are shown in green.

Amplitude The cumulated amplitude of the visual motion flow for the previous experiment is shown in Fig.4.12. Each spatial location is associated with its estimated flow amplitude after a single rotation. As the computed component of the flow are inverted, high amplitudes correspond to slow motion, and vice versa. As expected, the velocity of the rotating bar increases with the radius. The expected theoretical ratio of the velocity between the outer and the inner rim is 3.09, the estimated ratio after computation is 3.07.

A sample of flows computed from the same sequence on the same time period

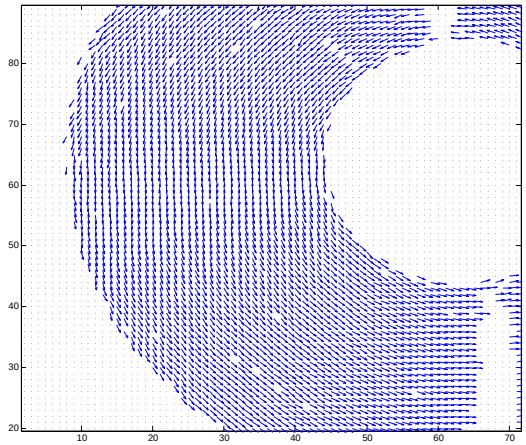


Figure 4.11: Sample of computed event-based flow for the stimulus in Fig.4.1 shown for a cumulated period of time of 3 ms for a portion of the focal plane for better readability.

.

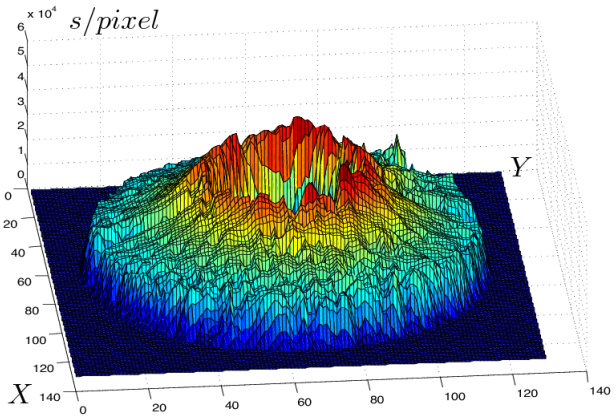
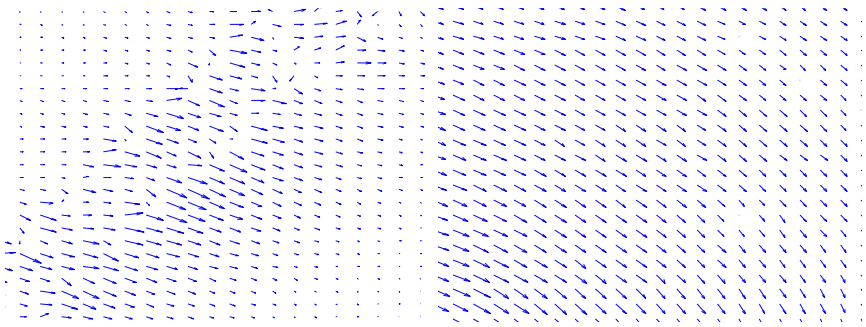


Figure 4.12: Amplitude of the computed motion flow for the stimulus in Fig.4.1.

of 3ms for frame-based and event-based methods is shown in Fig.4.13. Frames have been generated by cumulating events to simulate a frame rate of 3ms to ensure clarity in the display of results and allow comparison on a short time scale. In absence of a fast camera, and due to the simplicity of the stimulus, cumulating events has shown to efficiently approximate frames. In Fig.4.13(a), frame-based computation produces a wide variety of amplitudes responses due to the static nature

of frame-based acquisition: it is usually a non smooth vector field because of the discontinuities introduced by frames' sampling. Proposed methods to improve the smoothness of the vector field(55) apply a general smoothing paradigm that induce additional artifacts. Computed event-based amplitudes in Fig.4.13(b) are smooth and closely correspond to the expected amplitudes thanks to the high temporal resolution of the DVS.



(a) (b)

Figure 4.13: (a) Frame-based optical flow using Horn-Schunk on a 3ms accumulation frame. (b) Accumulation of the Event-based optical flow over a same duration of 3ms.

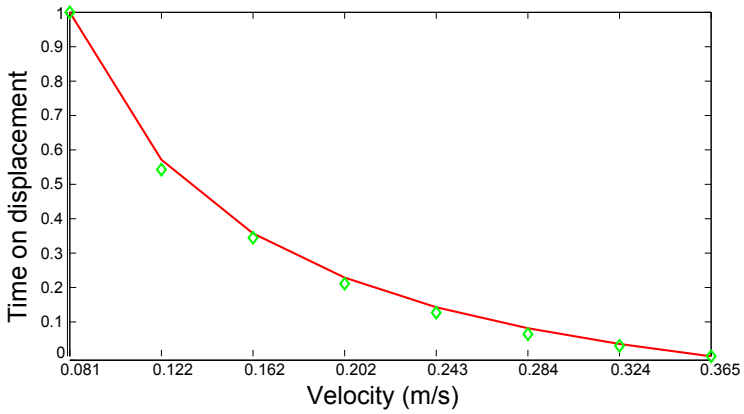


Figure 4.14: The estimated amplitude (green diamond) is shown with the real one (red line).

The amplitudes of the visual flow are computed for different known belt speeds

obtained by setting the DC motor of the setup shown in figure 4.2 with a range of velocities of $0.081\text{m} \cdot \text{s}^{-1}$ to $0.365\text{m} \cdot \text{s}^{-1}$. In order to compare the real and the computed amplitudes, we normalize the estimated amplitudes and the velocities of the moving patterns with their respective maximum values. The normalized results, shown in Fig. 4.14, correspond to the estimated velocity of the motion flow and the ground truth ranges in the interval $[0.081, 0.365]\text{m} \cdot \text{s}^{-1}$. The ground truth and the estimated values coincide with a mean error of 9%.

Computational time We define Δt_b as the time interval within which the visual signal is observed, it is equal to the inverse of the frame-rate of the standard perspective camera (i.e. $\Delta t_b \sim 33ms$ if the camera captures at 30fps). For a meaningful comparison, this time duration is chosen to evaluate the frame-based and the event-based techniques. Let Δt_c be the processing time consumed for the flow estimation from the signal acquired during Δt_b. We then define the ratio r such as: $r = \frac{\Delta t_c}{\Delta t_b}$. If $r \leq 1$ then the computation can be performed in real-time.
For the experiments, the computation of the frame-based optical-flow uses the OPEN-CV implementation Lucas and Kanade's (76) algorithm. It relies on a pre-processing of incoming images to select features (77) on which the motion visual motion flow is then estimated. Fig.4.15(a) and Fig.4.15(c) show the number of events and features used to compute the flow for the rotating bar. As illustrated in Fig.4.15(b), the maximum ratio in the case of the event-based algorithm is $r = 0.1$, showing that the visual flow is estimated in real-time. The frame-based implementation processing time ratio is $r = 0.0395$ as shown in Fig. 4.15(d). It allows also real-time estimation, assuming only a few image features are detected.

The mean number of events and the mean processing ratio allow to estimate the mean processing time of a single event: $33 \times 0.1/60 \sim 0.055$ms (mean number

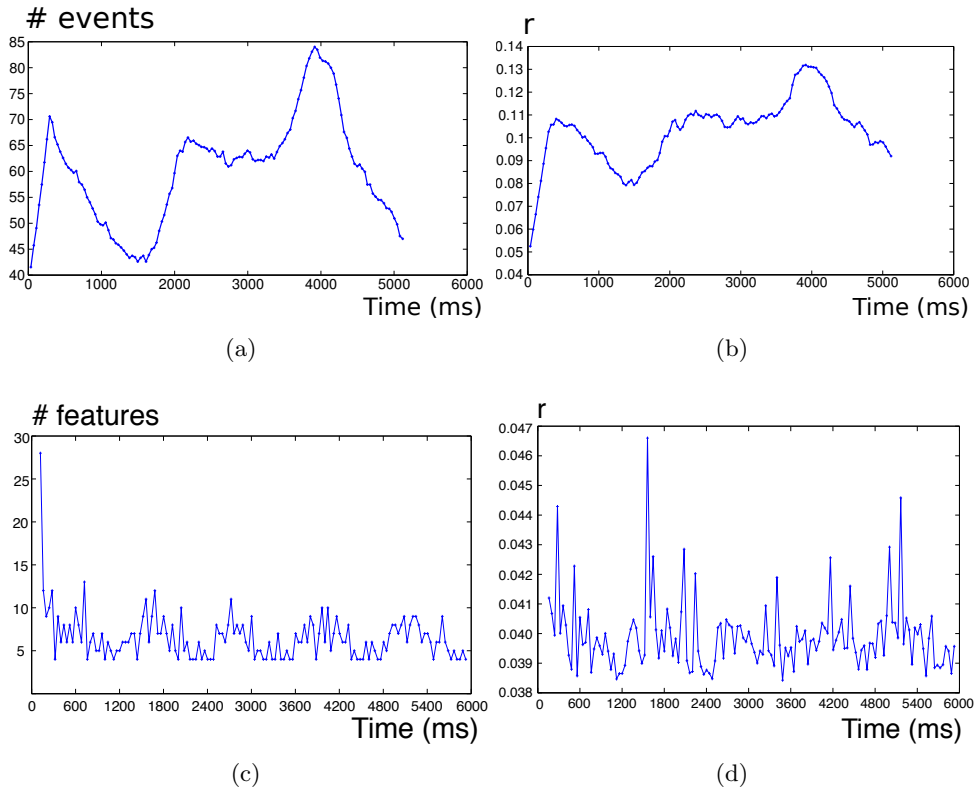


Figure 4.15: (a) Number of events per time bins generated by the rotating bar shown in Fig.4.1. (b) Processing time ratio of the event-based technique: the mean processing ratio is equal to 0.1, for a mean number of events equal to 60. (c) Number of features using cumulated frames generated by the optimized Lucas-Kanade's algorithm. The algorithm selects the feature pixels on which the optical flow is computed. (b) Processing time ratio of the Lucas-Kanade's technique: the mean processing ratio is equal to 0.0395 and the mean number of features is around 6.

of events per millisecond = 60 and mean processing ratio r= 0.1). With the same consideration, the mean time to process one image feature with the frame-based optimized implementation is equal to $33 \times 0.0395/6 \sim 0.217$ms (mean number of feature per millisecond = 6 and mean processing ratio per feature r=0.035). This means that within the time slot of 33ms, around 600 events can be processed in real-time with the first technique, while only 150 pixels can be processed in real-time in the second technique. The Lucas-Kanade C++ OpenCV implementation is highly optimized if compared to the C++ implementation of the event-based

algorithm. Presented results can then be seen as a lower bound of the event-based algorithm performances. Event-based motion flow is expected to run even faster with an adequate algorithm optimization, since its computational principle is very simple.

Natural Scenes In the case of natural scenes, the flow estimation is harder to evaluate as the ground truth is not available. However, it is still possible to show the coherence of the computed flows with the scene's content. In the first set of data shown in Fig. 4.16 (a-d), the velocity vectors amplitudes of cars moving along highway lanes using the event-based sensor are shown Fig. 4.17 (e-h). Velocities are increasing as the cars are getting closer to the sensor, this is a direct effect of the perspective projection Fig. 4.18 (i). A second sequence acquired by the sensor is shown in Fig. 4.19 4.20 4.21. It confirms the same observations. Again as expected, the velocity amplitude increases when the distance to the sensor decreases.

4.1.3 Limitations

The sensor is subject to several non-idealities that cause latencies in the signal acquisition, limiting the accuracy of the computed flow. In the following experiment we consider a rotating moving bar, as shown in Fig. 4.21. The bar observed by both the spiking sensor and a conventional camera has an angular speed increasing progressively from 450 to 5000 rpm.

The first column (a,e,i,m,q) shows samples of images acquired by a conventional camera at 30fps. The second column (b,f,j,n,r) shows the results using the Horn and Schunk's algorithm. As expected frames are not suitable for these high speed motion as the motion blur is preventing flow estimation, motion results are chaotic and irregular and not fitting the general rotational motion of the bar. The motion flow is already inaccurate at the lowest speed as shown by Fig. 4.21(b).

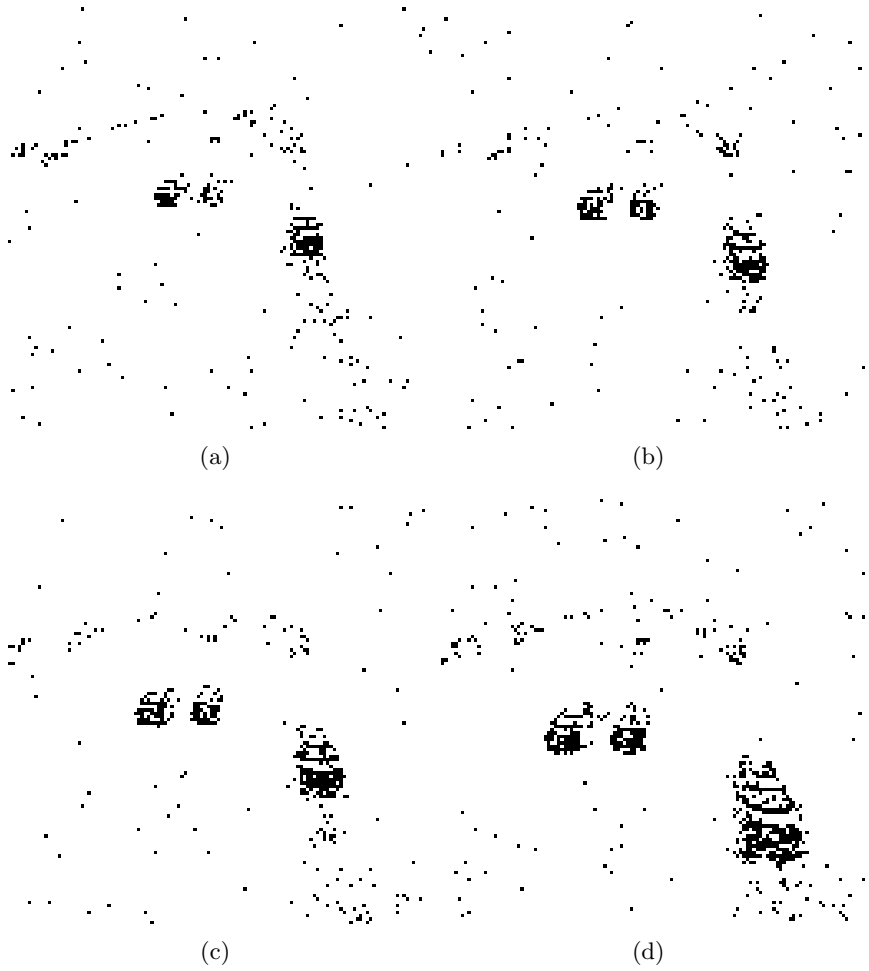


Figure 4.16: (a-d) Outdoor scene acquired by the event-based retina showing cars on a highway using cumulated events on the focal plane.

The third column (c,g,k,o,s) shows the events generated by the rotating bar in the spatiotemporal space of events. As the bar rotates faster, more events are collected and longer portions of the motion are acquired. The last column (d,h,l,p,t) provides the flow estimation using the event-based algorithm. The flows are accurately estimated in real-time for all rotational speeds. As the rotational speed increases, the motion flow in the event-based case tends to be sparser, Fig.4.22 shows the motion blurs, resulting from the retina's latencies when capturing light. This is the main

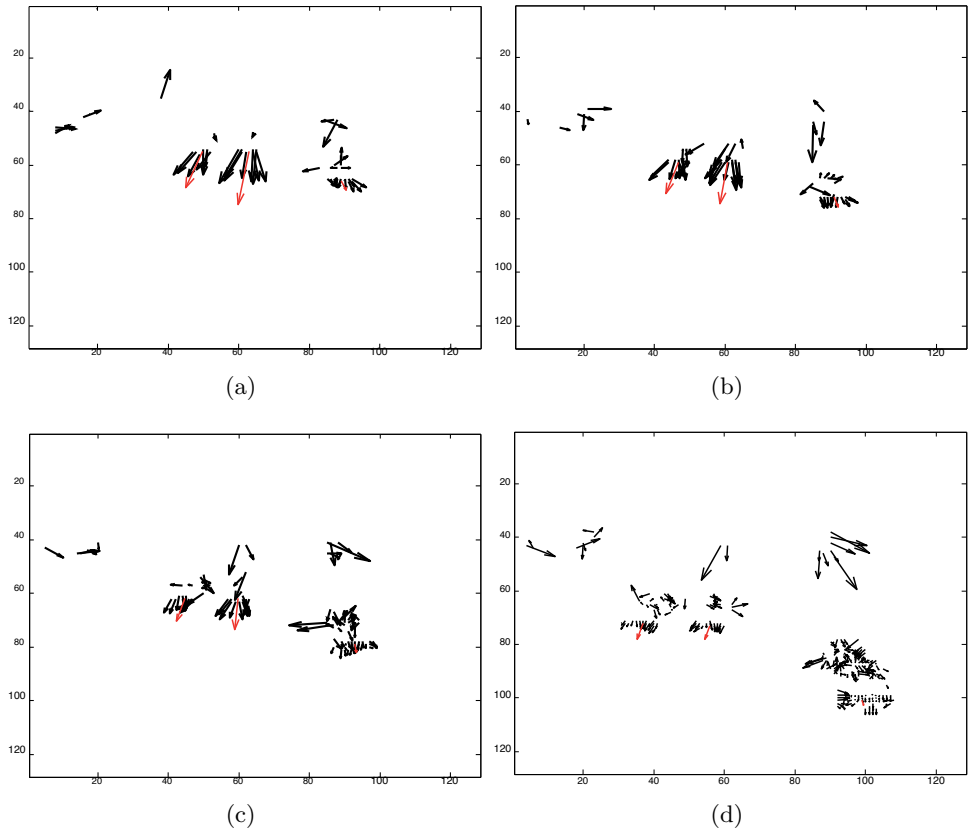


Figure 4.17: (e-h) show the mean motion flow for each vehicle (arrows) increasing as the cars get closer to the retina.

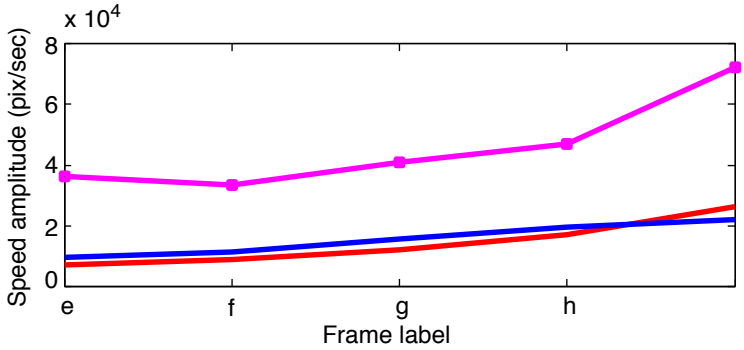


Figure 4.18: Mean velocities amplitudes shown in (i) for each of the three cars of Fig 4.17(e-h).

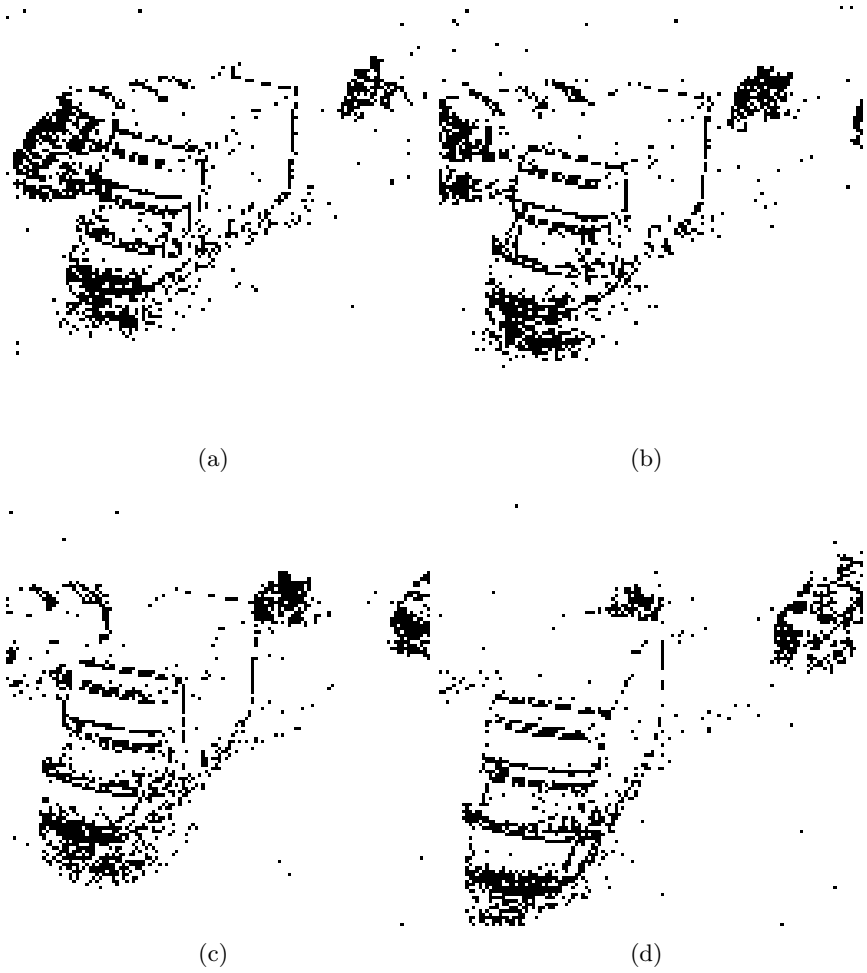


Figure 4.19: (a-d), Cumulated events on the focal plane of an outdoor traffic scene.

limiting element of the retina's in estimating accurately the flow. With the increasing speed of the rotation, motion blurs induces clusters of events instead of sharp edges. The retina is also not generating a sufficient amount of events form all spatial locations, some pixels are not activated by the moving bar. The fitting of Σ_e is then inevitably affected. As shown in Fig.4.23, the estimated amplitude and orientation of the bar follow closely the ground-truth up to a rotation velocity of 2500 rpm. Beyond this velocity, the orientation seems to stay stable up to 4000 rpm. Less events are collected at high speed, thus affecting the slope of fitted plane but not its

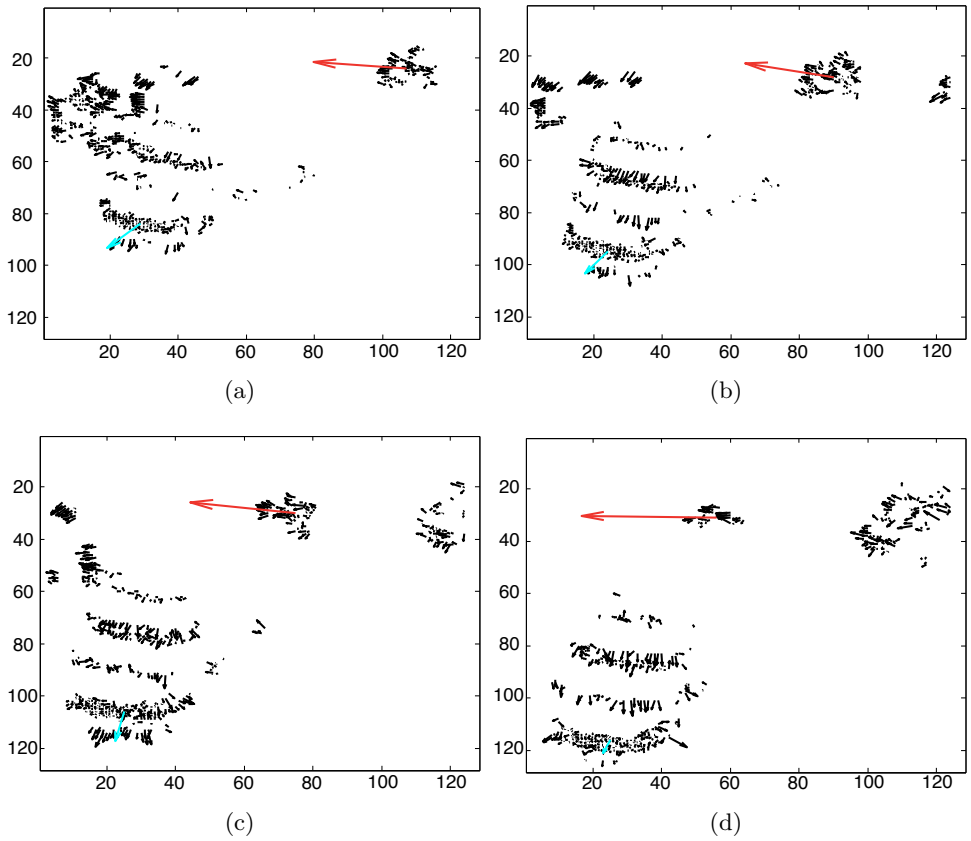


Figure 4.20: In (c-h) the car moving from left to right (in the background) has an almost constant depth. The truck moving toward the retina is showing an increasing velocity amplitude due the perspective projection.

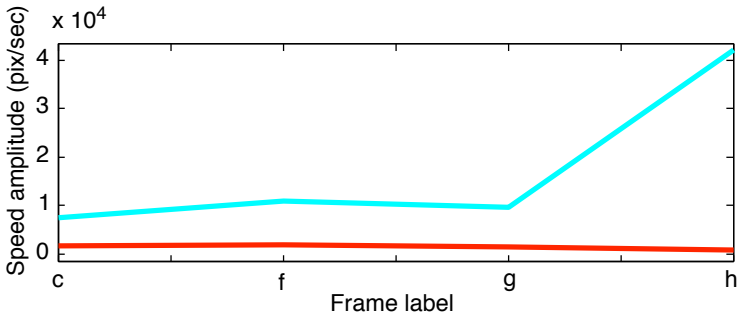
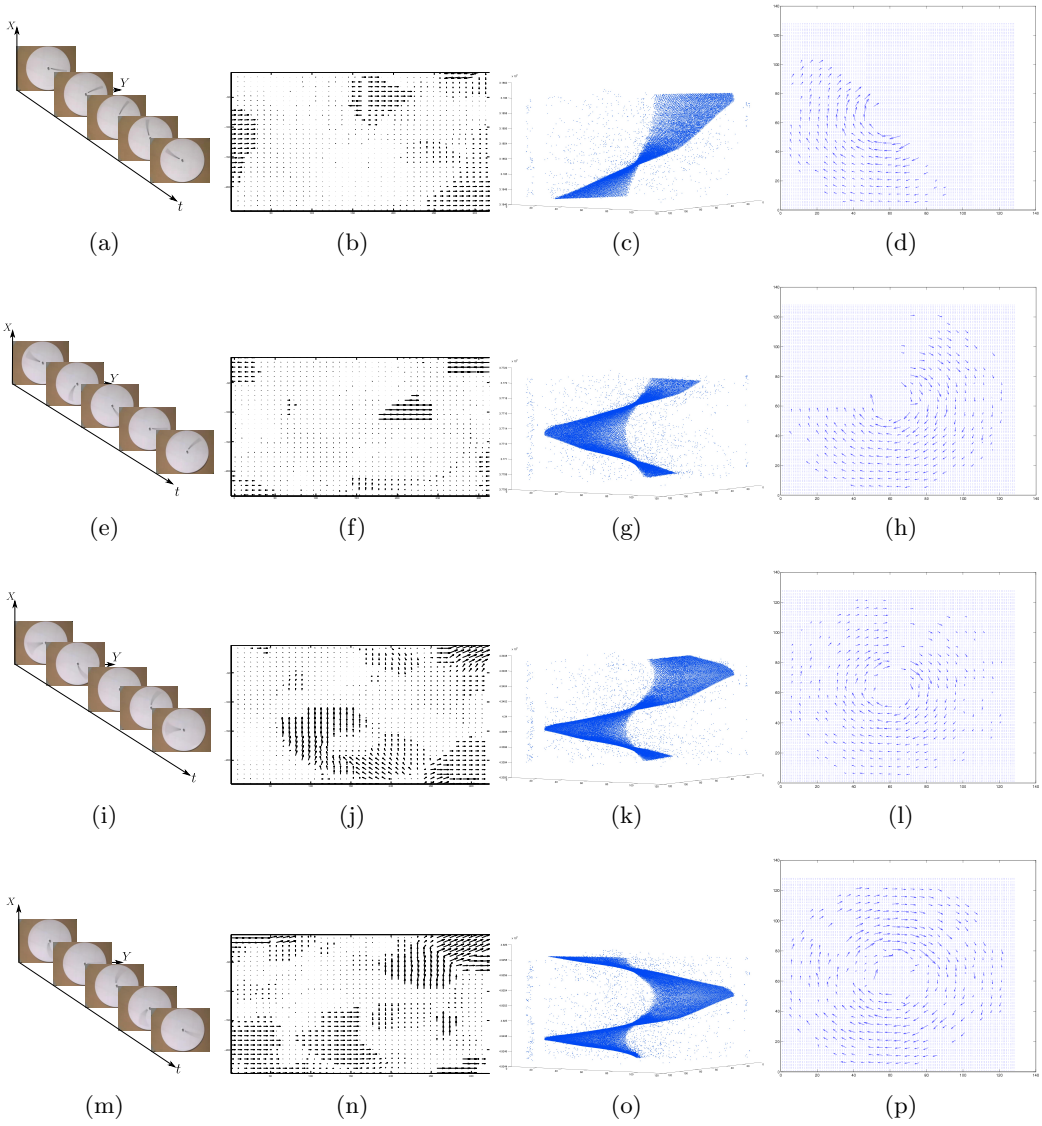


Figure 4.21: In (i) the mean velocities amplitude of each vehicle is shown: bottom curve is reflecting the almost constant velocities of the car, while the increasing one provides the velocity of the truck moving toward the retina.



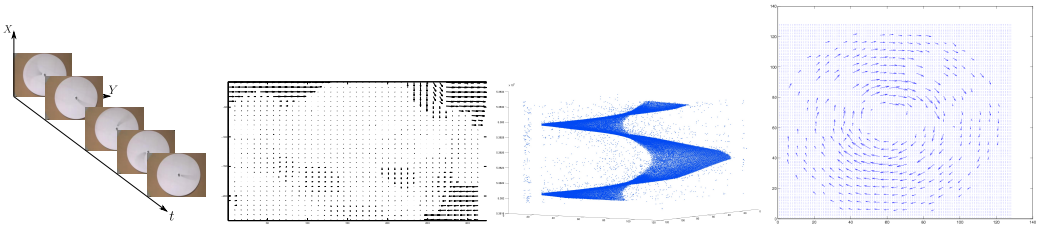
(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

direction. This is the limit of the sensor. The use of better lighting conditions will lower these effects.



(q) (r) (s) (t)

Figure 4.21: Comparison of the flows computed with a 30fps frame-based camera and the DVS. The disk is rotated at a speed ranging from 450 to 5000 rpm with an elementary step of 500 rpm. The first column (a,e,i,m,q) shows samples of images acquired by a conventional camera at 30fps. The second column (b,f,j,n,r) shows the results using the Horn and Schunk's algorithm. As expected frames are not suitable for these high speed motion as the motion blur is preventing flow estimation, motion results are chaotic and irregular and not fitting the general rotational motion of the bar. The third column (c,g,k,o,s) shows the events generated by the rotating bar in the spatio-temporal space of events. One can notice that as the bar rotates faster, more events are collected and longer portions of the motion are acquired. The last column (d,h,l,p,t) provides the flow estimation using the event-based algorithm, the flows are accurately estimated in real-time for all rotational speeds.

4.2 Focus of expansion and Time to contact

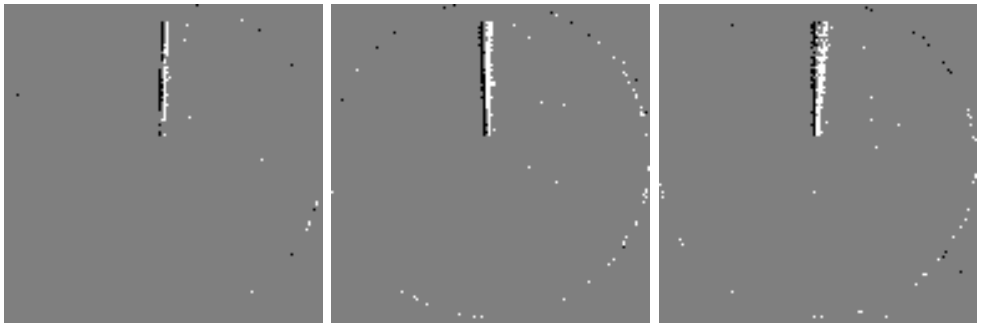
In this section I present the results of the different methods explained in sections 3.2 and 3.3. I first illustrate the accuracy and efficiency of the computation of the FOE using synthetic and real data. In a second stage the results of the TTC algorithm are shown on real data.

The real data are all acquired using a *Pioneer 2* mobile robot, Fig.4.24a, equipped with a DVS sensor, Fig4.24c, and a laser range finder (LRF), Fig4.24b.

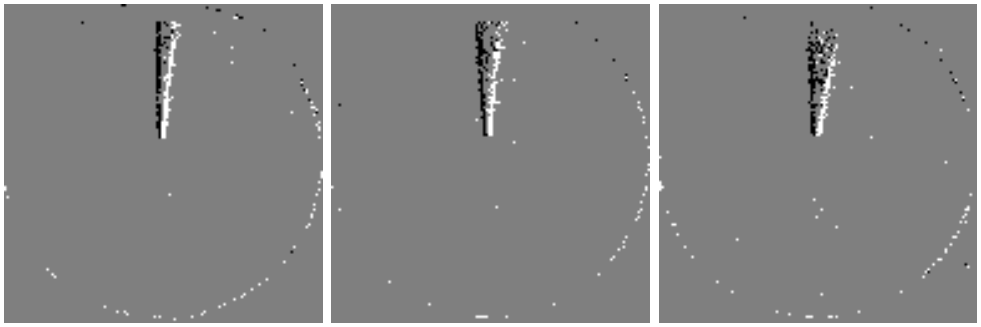
The acquisition with camera are done in normal indoor luminance. All computations take place on a *intel*® core2 duo laptop under Linux.

4.3 Focus of expansion

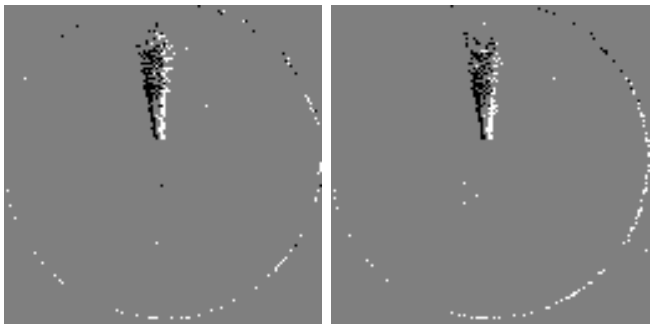
Using the event-based visual flow computation presented section3.1.3, we are able to compute efficiently and accurately the FOE over continuous time. In the following



(a) 405 $tr.min^{-1}$ (b) 1255 $tr.min^{-1}$ (c) 1870 $tr.min^{-1}$



(d) 2450 $tr.min^{-1}$ (e) 3330 $tr.min^{-1}$ (f) 4110 $tr.min^{-1}$



(g) 4530 $tr.min^{-1}$ (h) 4840 $tr.min^{-1}$

Figure 4.22: Sequences of a disk rotating with an increasing speed from 450 rpm to 5000 rpm for the event-based retina. Cumulated events in the focal planes are shown for a fixed time period of 500ms. Events underlining the edges are more and more scattered as the rotation speed increases, this is a typical effect of the motion blur.

we present three experiments to demonstrate this statement.

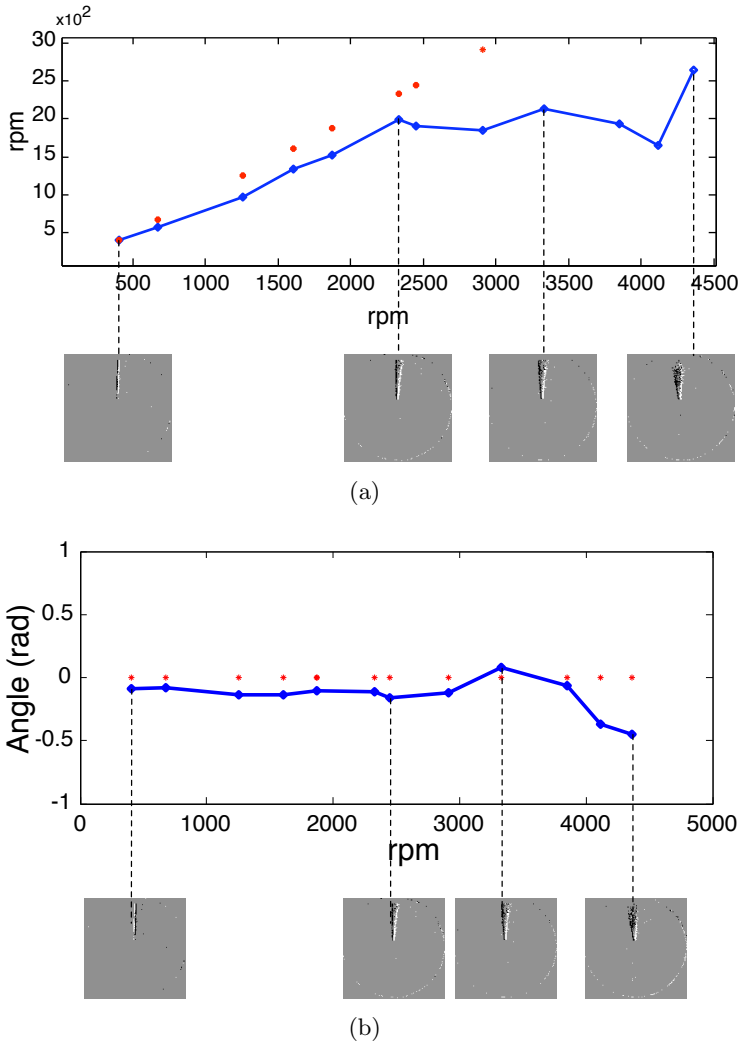


Figure 4.23: In (a) the mean velocity of the estimated flow and the mean angle estimation in (b) are shown using the event-based retina from the event flow of the rotating bar. The angular speed increases from 450 to 5000 rpm. Estimations are given each time the bar reaches its initial starting position. The red dots provide the ground-truth. The estimation of the motion parameters is stable up to 2500 rpm. Beyond the orientation seems to stay stable up to 400 rpm. Less events are collected at high speed thus affecting the slope of fitted plane but not its direction. This is the limit of the sensor for the used lighting conditions.

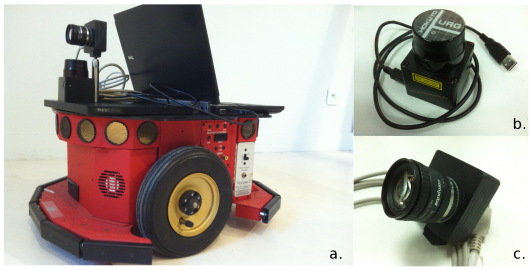


Figure 4.24: Illustration of the Pioneer 2, used for the following experiments, fig.4.24a, carrying a DVS asynchronous sensor, fig.4.24c, and a laser range finder, fig.4.24b.

4.3.1 Simulated data

In order to accurately evaluate our method using a ground a truth, we confront the algorithm to synthetics data. We first simulate the expansion of a disk as it would occur while a robot runs closer to a circular target. Figures 4.25 presents the results of the two components of the FOE, the horizontal axis on top, and the vertical axis on middle. In both subfigures we plot the ground truth as a continuous line, the raw data in dashed line and the data smoothed using a sliding windows of 1ms with a dotted line. The third subfigure presents the stability of the computed flow, which gives a confidence criteria on the computation. We simply define this stability criteria as the normalized 1-norm distance between the last and the current FOE.

Reconstructed frames, created using an accumulation over time of the events from the DVS, illustrate the movement of the circle while expanding. First, a stable focus of expansion arises from the computed motion flow, fig.4.25a, then a period of stability occurs while the movement continues to induce a dense apparent velocity around the edge of the circle, fig.4.25b. This stable focus of expansion is computed until the end of the simulation, fig.4.25c.

To smooth variations of the computed FOE we apply a convolution using a sliding window over $1ms$. The similarity between the raw and convolved data of the algorithm and the ground truth is computed using the mean square error, the results are shown tab.4.1. The ground truth of the FOE is centered in $(64, 64)$, respectively

	Mean square error		Standard deviation	
	Horizontal	Vertical	Horizontal	Vertical
Raw data	$5.44 pixels^2$	$4.07 pixels^2$	$9.0 pixels^2$	$7.65 pixels^2$
Convolved data	$1.69 pixels^2$	$0.96 pixels^2$	$1.35 pixels^2$	$1.2 pixels^2$

Table 4.1: Results of similarity for the simulated illustrated 4.25a-c. The similarity is given by the computation of the mean squared error between the ground truth and the raw results and the convolved results of the TTC.

on the x and y coordinates.

As the results show, the similarity with the ground truth is increased by the smoothing of the data. It is noticeable by the decrease of the mean square error and the standard deviations showing a better stability. This convolution can be applied incrementally to increase the precision of the computation in real conditions.

The computation of the FOE is possible as long as the motion involves an isotropic enough divergence of the optical flow field. This is the case if the direction's change of a robot is slower than its translation along the axis of the camera. To show the accuracy of the computation in this condition, we simulate a second synthetic expanding circle which center translates along the diagonal of the visual area. The result of the computation is given figure 4.26, where we use the same convention than in figure 4.25.

As above, we illustrate the experiment with reconstructed frames from the events of the camera. We observe that the computation of a stable FOE arises after few iterations, fig.4.26a, and is stable along the trajectory, fig.4.26b. At the end of the simulation, the point of divergence of the flow is lost out of the synthetic space, making impossible to accurately locate the FOE, fig.4.26c. The unreliable computation of the FOE is depicted by the high instability of our criteria, fig.4.26 bottom subplot.

The table 4.2 shows the similarities computed as above, where the ground truth is closely followed by the computed FOE. This accuracy to follow the shift of the

	Mean square error		Standard deviation	
	Horizontal	Vertical	Horizontal	Vertical
Raw data	$122.57 pixels^2$	$109.83 pixels^2$	$198.42 pixels^2$	$195.45 pixels^2$
Convolved data	$24.09 pixels^2$	$6.16 pixels^2$	$19.83 pixels^2$	$9.47 pixels^2$

Table 4.2: Results of similarity for the simulated illustrated 4.26a-c. The similarity is computed as above.

FOE shows the ability of the method to give the head direction of a robot which does not strictly follow a pure translational movement along the axis of the camera. As in the previous experiment the similarity is increased by convolving the data.

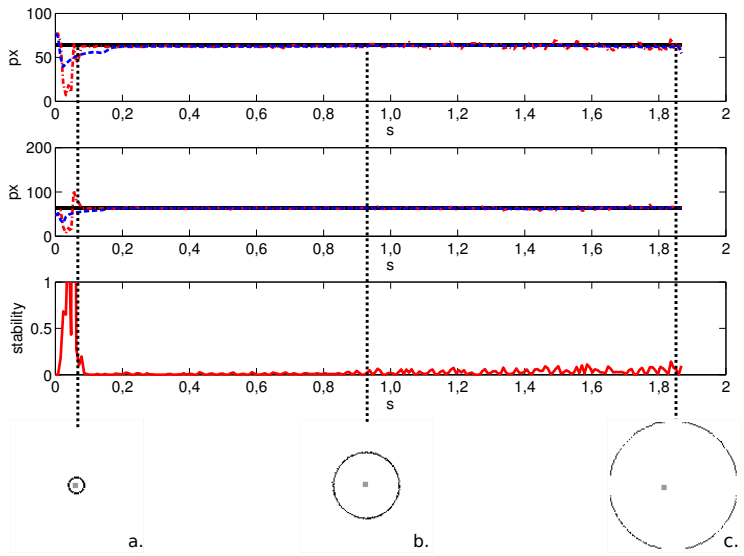


Figure 4.25: Simulated data Expansion of a synthetic circle simulating a displacement of a robot toward a circular target perpendicularly to the axis of the camera. The subplot present the results of the FOE computation for the horizontal, on top, and vertical, on bottom, component of the flow. The continuous lines represent the ground truth, while the dashed and dotted line represent the raw data and the data smoothed using a sliding window of 1ms. The third subplot present the stability of the FOE given the current and the previous results. The frame on bottom illustrate the state of the circle. We notice that the stable part of the plot corresponds to a well defined edge of the circle.

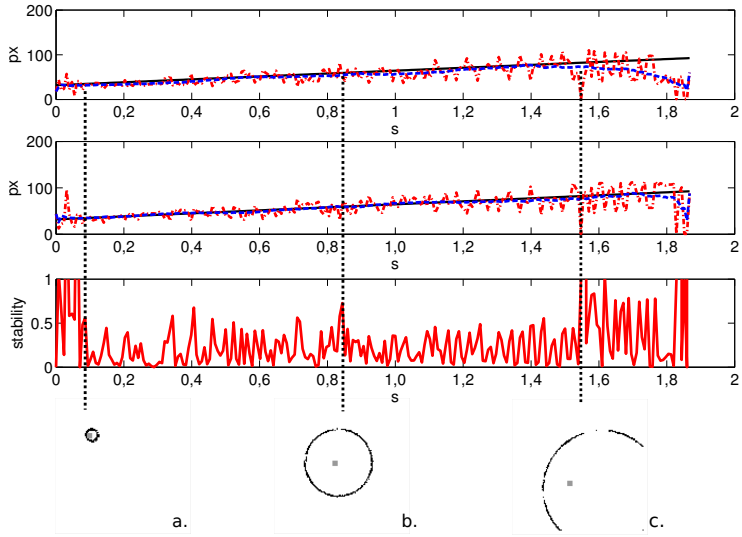


Figure 4.26: Simulated data Simulation of a robot moving forward a circular target not aligned with the axis of the camera. The convention of the illustration is the same than the figure 4.25.

	Standard deviation	
	Horizontal	Vertical
Raw data	$12.43px$	$12.67pixels^2$
Convolved data	$8.61pixels^2$	$6.43pixels^2$

Table 4.3: Results of standard deviation computation of the experiment illustrated fig.4.27.

4.3.2 Real data

A DVS camera is embedded on the mobile platform which runs along the axis of the camera in indoor environment. Figure 4.27 illustrates the results with the same convention than figures 4.25 and 4.26. The FOE is accurately computed from $t_a = 0.3s$, and is stable until the robot slows down and stops, $t_b = 1.84s$, where the FOE cannot be computed. As for the simulated data, the range in which the FOE is accurate is evaluated by the computation of the stability criteria.

To illustrate the stability of the retrieved egomotion, we calculate the standard deviations along the trajectory in the range surrounded by the stability criteria, $t \in [0.3, 1.75]s$. The results of the computation is shown tab.4.3.

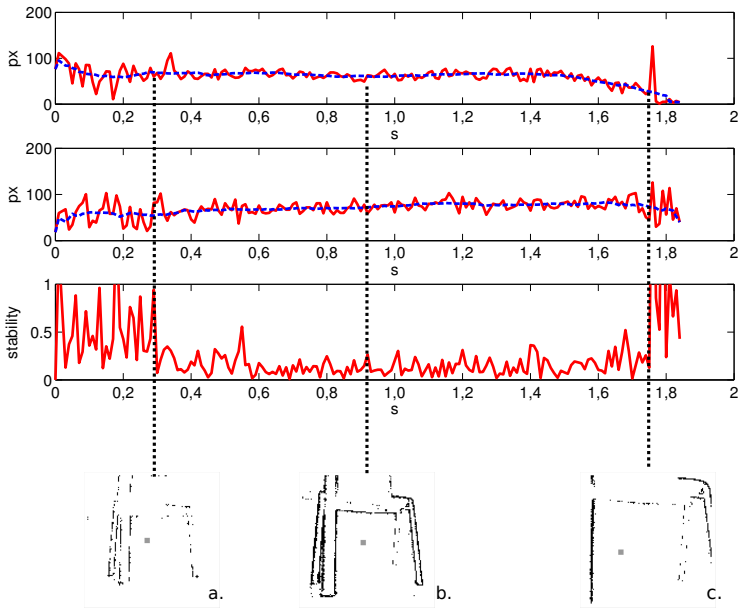


Figure 4.27: Real data Mobile platform moving toward a chair along the axis of the camera. The two subplots represent the two components of the FOE. The Dashed line illustrates the raw data, while the dotted line illustrates the data smoothed using a sliding windows of 1ms. The last subplot depicts the stability of the computation given the current and last FOE. The frames a-c, illustrating the motion, are reconstructed using an accumulation over time of the event of the DVS.

4.4 Time to contact

The time to contact is measured from the event-driven optical flow and compared to the results given by a Laser Range Finder embedded on the mobile platform. The LRF allows a continuous acquisition of the distance between the platform and the obstacles. The scans are timestamped, we are able to compute the velocity of the platform and the TTC.

For the aim of the experiment, the robot moves back and forth in direction of an obstacle perpendicularly to the axis of the camera, during which we record the data from the two sensors.

As we use an uncalibrated DVS, and the sensors extracting the information of the scene in different ways and in different fields of view, the results obtained are

not directly comparable. As in (72), we only consider the TTC correlated with the obstacle in front of the setup, Fig.4.28a and b. The similarity between the results is given by the computation of the cross correlation of the data, figure 4.28c, where the cross correlation is normalized by the LRF auto-correlation. In order to avoid obstacles we are interested in objects which involve a situation of risk. As it can be seen on the figure 4.28c, $t \in [1.2, 6]s$, the quantitative worst similarity in this case is 60%. This low quantitative similarity can be explain by:

- The use of uncalibrated sensors,
- false range measurements resulting when the laser beam reflects from more than one surface,
- sensor readings, which may be erroneous because of specular reflections.

For these reasons, it is more relevant to focus on qualitative comparison between the results4.28 a-b, where in this case the two sensors clearly provide the same information.

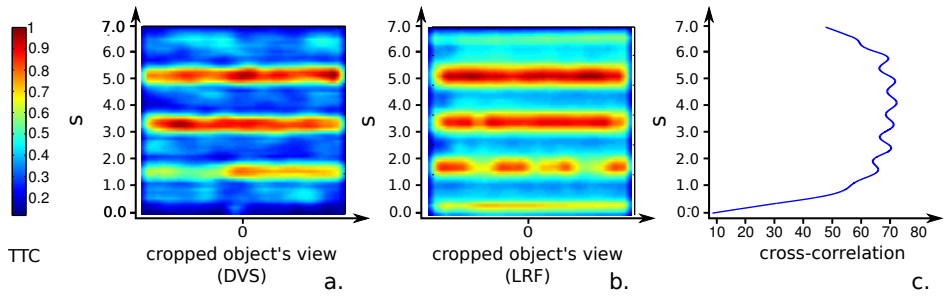


Figure 4.28: The time to contact computed using the DVS (a), and the laser range finder (b) are illustrated using a color map, which color scale express the imminence to impact in the selected central view of the sensors. The correlation between the results of the two sensors is depicted in (c), giving highlighting the similarity of the information provided by the two sensors, the laser range finder being considered as the ground truth.

Chapter 5

Conclusion

5.1 Asynchronous frameless event-based flow

This thesis introduced a complete framework to use event-based accurately timed information for visual processing for robot navigation. The developed approached focused on the estimation of the optical flow. Results show that event-based computation introduces a paradigm shift in visual computation. The use of accurately timed events allow unprecedented high speed computation at the lowest cost.

This contrasts with current main stream approaches that rely on images. The work developed in this project followed two constraints, accuracy and efficiency (78),(79).

- no matter how accurate an algorithm may be, it is not useful unless it can output the results within the necessary response time for a given task,
- Even if it is fast, an algorithm is useless unless it computes motion sufficiently accurately (78).

The time oriented event-based computation paradigm allows to deal with this trade-off by using a new way of acquiring visual information from the real world. The

counter-part is an *a priori* less "human-understandable" information, because of the use of events rather than images. It is important to emphasize that images are unknown to biological vision system. They are the heritage of early painting and photography and are best suited for accurately reproducing the external world, rather than extracting information out of the sensor's data. Biological sensors are event-driven and precisely timed (1ms).

5.2 Visual flow

The study of the visual flow from events (specially the second approach introduced in section 3.1.3), shows that one can use the space of co-active events to directly derive information about the direction of a visual stimulus. The precise timing conveyed by the Neuromorphic asynchronous event-based vision sensor is fully used to determine locally for each incoming event its direction and amount of motion. Timing is the essential computational element, the whole computation is based on its precision. The method complies with the concepts of event-based computation and the processing is performed on each incoming event rather than on a time interval. The presented work differs from existing frame-based techniques that consider temporal window frames ($\sim 33 - 500$ms) and induce unnatural high computational costs. The developed method can be applied equally on other modalities such as the tactile one that shares common characteristics with vision. Both rely on a spatial grid of sensors (pressure: mechano-receptor, light: photoreceptor) that input to a chain of processing precise neural responses.

The presented approach did not impose any mathematical model, results show that the intrinsic properties of spatiotemporal spaces provide the inverse of the velocity. This observation sets the estimation of motion on time rather than space. The precise timing properties of the method comply with the dynamic properties of natural

environments and the importance of time measurement in the brain; actions are driven by time, motor control depends on processes that have to determine when exactly to perform actions. This then adds to the ongoing debate around whether perception and movement share common time lines or are organized on separate clocks and coding. The presented results suggest that the same timing mechanism identified at the perception level may also underlie motion behavior. Event-based acquisition provides a fast common mechanism that allows perception to be linked to motion by a common computationally inexpensive timing system. This could then provide an efficient way of controlling sensory dependent behavior and anticipating changes in the environment.

5.3 Focus of expansion and Time to contact

The focus of expansion and the time to contact computation presented section 3.2 and 3.3 rely on the motion flow low computational costs. The approach developed here fulfills the requirements of event-driven computation, it is incremental, in the sense that events are processed at the exact time of their arrival. There is no need to store "frames" as a collection of events to compute the time-to-contact. This contrast again with main stream methods that face limitations due to the slow computation of the optical flow. This work is in adequacy with biological findings of Lee (58), the fundamental ecological stimulus for vision is not a camera like static time-frozen image, rather a constantly changing optic array or flow field, the description of which must be in spatio-temporal terms, where one can directly set the link with the event based paradigm. By the accurate computation of the focus of expansion and the time to contact based on biological approach of precise timing, we showed that the frame-based paradigm is non physiological compatible. This opens up new methodologies of computation in direct link with current physiological studies of the

brain.

Chapter 6

Discussion

My Phd work focused on studying the advantages of event-based computation and the practical use of asynchronous event-driven sensors in mobile robotics. This coincided with the availability of the first fully usable event-based sensor (the Dynamic Vision System). The path and struggle to determine new methodologies was among the most exciting endeavor of my training. I am fully convinced that this will open up a new way to think artificial vision.

The hardest point in this new context was to develop a real new methodology that works on time rather than on intensity information in order to fully take advantage of asynchronous acquisition. The goal was always to avoid the temptation to stay in known approaches that are inappropriate for such data. The attempt of copying or directly translating mainstream methods of computer vision showed rapidly its limitations. The optical flow computation opened the problem of the efficient use of the information acquired. Even if the first attempt extending the Lucas and Kanade light consistency constraint was successful, the method showed rapidly its limits. The use of temporal spatiotemporal space setting aside light consistency proved to be more efficient and illustrates the type of computation that should performed on such data. Spatiotemporal spaces are usually studied in physics, to our knowledge

this the first true use of their properties in artificial vision. The use of spatiotemporal spaces introduces elegant formulations of visual problems. The work unveiled the properties on tangent spaces but there is more to be found in the space-time structure of events specially when the problem of features is dealt with. An encouraging observation of this work is that the developed methods fit biological findings and recordings as it has been shown along the manuscript.‘ There is still so much to do in this field, we are currently at the beginning of a new paradigm that will necessitate more exchange between physiology and engineering. This has already started all over the world, the growth of the neuromorphic community is a sign that this will sooner of later happen and generalize. The other challenge that needs to be tackled is to rethink computation and get away from VonNeumann architectures and computational hardware that are totally inadequate to handle event-based massively parallel computation. Current computers cannot deal with this type of information, their architecture is too limiting as information is serialized to fit within the sequential processing scheme. New computation biomorphic-like hardware is being developed around the world such as : neural processors on chip(80), or highly parallel platforms like the spinnaker (81), but there is still a missing computational element. These new computer are currently used to simulate neurons and physiological findings, but there more to do. There is a need to developed and rethink computation beyond Turing machines computation-like to go toward more interactive computation. Unfortunately this will no happen until our current knowledge of the mathematics of the brain increases. It is for now still too scarce, brains deal with non linearities, and compute in a complete different level of understanding of our current knowledge. Perhaps the solution lays in path initiated by Godël where he describes a philosophical path from the incompleteness theorems to Husserl?s phenomenology and his investigation of the treatment of categories to think the future of computation.

References

[1] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori. **The iCub humanoid robot: an open platform for research in embodied cognition**. In *PerMIS: Performance Metrics for Intelligent Systems Workshop*, Washington DC, USA, August 2008. 4

[2] Willow Garage. **Overview of the PR2 robot**. http://www.willowgarage.com/pages/robots/pr2-overview. 4

[3] Misha Mahowald. **Analog VLSI Chip for Stereocorrespondence**. In *ISCAS*, pages 347–350, 1994. 6

[4] K. A. Zaghloul and K. Boahen. **Optic nerve signals in a neuromorphic chip II: testing and results**. *Biomedical Engineering, IEEE Transactions on*, **51**(4):667–675, 2004. 6

[5] P.F. Ruedi, P. Heim, F. Kaess, E. Grenet, F. Heitger, P.Y. Burgi, S. Gyger, and P. Nussbaum. **A 128/spl times/128 pixel 120-dB dynamic-range vision-sensor chip for image contrast and orientation extraction**. *Solid-State Circuits, IEEE Journal of*, **38**(12):2325–2333, 2003. 6

[6] U. Mallik, M. Clapp, E. Choi, G. Cauwenberghs, and R. Etienne-Cummings. **Temporal change threshold detection image**. *IEEE ISSCC*, pages 362–363, 2005. 7

[7] Eugenio Culurciello and Andreas G. Andreou. **CMOS image sensors for sensor networks**. *Analog Integrated Circuits and Signal Processing*, **49**(1):39–51, October 2006. 7

[8] P. Lichtsteiner, C. Posch, and T. Delbruck. **A 128 x 128 120 dB 15 us Latency Asynchronous Temporal Contrast Vision Sensor**. *Solid-State Circuits, IEEE Journal of*, **43**(2):566 – 576, Feb 2008. 7, 8, 20

[9] et al. Posch C. **High-DR Frame-Free PWM Imaging with asynchronous AER Intensity Encoding and Focal-Plane Temporal Redundancy Suppression**. *ISCAS*, 2010. 8

[10] Xiaochuan Guo, Xin Qi, and J.G. Harris. **A Time-to-First-Spike CMOS Image Sensor**. *Sensors Journal, IEEE*, **7**(8):1165 –1175, aug. 2007. 8, 13

[11] JA Leñero-Bardallo, T Serrano-Gotarredona, and B Linares-Barranco. **A signed spatial contrast event spike retina chip**. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 2438–2441. IEEE, 2010. 9

[12] Dongsoo Kim and Eugenio Culurciello. **A compact-pixel tri-mode vision sensor**. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 2434–2437. IEEE, 2010.

[13] T Delbruck and P Lichtsteiner. **Fast sensory motor control based on event-based hybrid neuromorphic-procedural system**. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pages 845–848. IEEE, 2007. 9

[14] J. Conradt, M. Cook, R. Berner, P. Lichtsteiner, RJ. Douglas, and T. Delbruck. **A pencil balancing Robot using a pair of AER dynamic vision sensors**. In *International Conference on Circuits and Systems*, 2009. 9

[15] Zhenjiang Ni, Cécile Pacoret, Ryad Benosman, Siohoi Ieng, , and Stéphane Régnier. **Asynchronous Event Based High Speed Vision for Micro-particles Tracking**. *Journal of microscopy*, 2011. 9

[16] Zhenjiang Ni, Aude Bolopion, Joël Agnus, Ryad Benosman, and Stéphane Régnier. **Asynchronous Event-Based Visual Shape Tracking for Stable Haptic Feedback in Microrobotics**. 9

[17] P. Rogister, R. Benosman, S.H. Ieng, P. Lichtsteiner, and T. Delbruck. **Asynchronous Event-based Binocular Stereo Matching**. *IEEE Transactions on Neural Networks*, 2011. 9

[18] S. Thrun, M. Bennewitz, Burgard M., W. Cremers, F. Dellaer, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. **MINERVA: a second-generation museum tour- guide robot. In Robotics and Automation**. *Proceedings. 1999 IEEE International Conference on*, **3**:1999–2005, 1999. 10

[19] R. Manduchi, A. Castano, A. Talukder, and L. Matthies. **Obstacle detection and terrain classification for autonomous off-road navigation**. *Autonomous Robot*, **18**:81–102, 2005. 10

[20] A. Stentz, A. Kelly, P. Rander, H. Herman, O O. Amidi, R. Mandelbaum, G. Salgian, and J. Pedersen. **Real-time, multi-perspective perception for unmanned ground vehicles**. *AUVSI*, 2003. 10

[21] J. Ibanez-Guzman, X. Jian, A. Malcolm, Z. Gong, C. Chan A., and Tay. **Autonomous armoured logistics carrier for natural environ-**

ments. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, **1**:473–478, 2004. 10

[22] M. Batalin, G. Shukhatme, and M. Hattig. **Mobile robot navigation using a sensor network**. *IEEE International Conference on Robotics and Automation (ICRA)*, 2003. 10

[23] Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver. **Visual navigation for mobile robots: a survey**. *Journal of Intelligent & Robotic Systems*, **53**(3):263–296, 2008. 11

[24] Andrew J Davison and Nobuyuki Kita. **Sequential localisation and map-building for real-time computer vision and robotics**. *Robotics and Autonomous Systems*, **36**(4):171–183, 2001. 11

[25] Robert Sim and Gregory Dudek. **Learning generative models of scene features**. *International Journal of Computer Vision*, **60**(1):45–61, 2004. 11

[26] Niall Winters and José Santos-Victor. **Omni-directional visual navigation**. In *Proceedings of the 7th International Symposium on Intelligent Robotics Systems*, pages 109–118. Citeseer, 1999. 11

[27] José Gaspar, Niall Winters, and José Santos-Victor. **Vision-based navigation and environmental representations with an omnidirectional camera**. *Robotics and Automation, IEEE Transactions on*, **16**(6):890–898, 2000. 11

[28] Sumit Badal, Srinivas Ravela, Bruce Draper, and Allen Hanson. **A practical obstacle detection and avoidance system**. In *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, pages 97–104. IEEE, 1994. 11

[29] R Gartshore, A Aguado, and C Galambos. **Incremental map building using an occupancy grid for an autonomous monocular robot**. In *Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on*, **2**, pages 613–618. IEEE, 2002. 11

[30] Nick Pears and Bojian Liang. **Ground plane segmentation for mobile robot visual navigation**. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, **3**, pages 1513–1518. IEEE, 2001. 12

[31] Liana M Lorigo, Rodney A Brooks, and WEL Grimsou. **Visually-guided obstacle avoidance in unstructured environments**. In *Intelligent Robots and Systems, 1997. IROS'97., Proceedings of the 1997 IEEE/RSJ International Conference on*, **1**, pages 373–379. IEEE, 1997. 12

[32] Miin Tyi Chao, Thomas Braunl, and Anthony Zaknich. **Visually-guided obstacle avoidance**. In *Neural Information Processing, 1999. Proceedings. ICONIP'99. 6th International Conference on*, **2**, pages 650–655. IEEE, 1999. 12

[33] Sjoerd van der Zwaan and José Santos-Victor. **An insect inspired visual sensor for the autonomous navigation of a mobile robot**. *Proc. of the Seventh International Sysposium on Intelligent Robotic Systems (SIRS)*, 1999. 12

[34] Ted Camus, David Coombs, Martin Herman, and Tsai-Hong Hong. **Real-time single-workstation obstacle avoidance using only wide-field flow divergence**. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, **3**, pages 323–330. IEEE, 1996. 12

[35] J Santos-Victor and Giulio Sandini. **Visual-based obstacle detection: a purposive approach using the normal ow**. In *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS), Karlsruhe, Germany*. Citeseer, 1995. 12

[36] Ted Camus. **Calculating Time-to-Contact Using Real-Time Quantized Optical Flow**. In *National Institute of Standards and Technology NISTIR 5609*, 1995. 12, 39

[37] James Crowley. **Navigation for an intelligent mobile robot**. *Robotics and Automation, IEEE Journal of*, **1**(1):31–41, 1985. 12

[38] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. **A 128×128 120 dB 15μs Latency Asynchronous Temporal Contrast Vision Sensor**. *IEEE Journal of Solid-State Circuits*, **43**(2):566–576, 2008. 12

[39] C. Posch, D. Matolin, and R. Wohlgenannt. **A QVGA 143 dB Dynamic Range Frame-Free PWM Image Sensor With Lossless Pixel-Level Video Compression and Time-Domain CDS**. *Solid-State Circuits, IEEE Journal of*, **46**(1):259 –275, jan. 2011. 12

[40] J.A. Lenero-Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco. **A 3.6 μs Latency Asynchronous Frame-Free Event-Driven Dynamic-Vision-Sensor**. *Solid-State Circuits, IEEE Journal of*, **46**(6):1443 –1455, june 2011. 13

[41] J.A. Perez-Carrasco, C. Serrano, B. Acha, T. Serrano-Gotarredona, and B. Linares-Barranco. **Event based vision sensing and processing**. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1392 –1395, oct. 2008. 13

[42] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gomez-Rodriguez, L. Camunas-Mesa, R. Berner, M. Rivas-Perez, T. Delbruck, Shih-Chii Liu, R. Douglas, P. Hafliger, G. Jimenez-Moreno, A.C. Ballcels, T. Serrano-Gotarredona, A.J. Acosta-Jimenez, and B. Linares-Barranco. **CAVIAR: A 45k Neuron, 5M Synapse, 12G Connects/s AER Hardware Sensory System for High-Speed Visual Object Recognition and Tracking**. *Neural Networks, IEEE Transactions on*, **20**(9):1417 –1438, sept. 2009. 13

[43] James J Gibson. **The ecological approach to the visual perception of pictures**. *Leonardo*, **11**(3):227–235, 1978. 13, 36

[44] Andrew B Watson and Albert Ahumada. *A look at motion in the frequency domain*, **84352**. National Aeronautics and Space Administration, Ames Research Center, 1983. 28

[45] D J Heeger. **Optical flow using spatiotemporal filters**. *International Journal of Computer Vision*, **1**:279–302, 1998. 28

[46] D J Heeger. **Model for the extraction of image flow**. *Journal of the Optical Society of America*, **4**:1455–71, 1987. 28

[47] J L Barron, D J Fleet, and S S Beauchemin. **Performance of optical flow techniques**. *International Journal of Computer Vision*, **12(1)**:43–77, 1994. 28, 49

[48] Padmanabhan Anandan. **A computational framework and an algorithm for the measurement of visual motion**. *International Journal of Computer Vision*, **2**(3):283–310, 1989. 28

[49] R Kories and G Zimmerman. **A versatile method for the estimation of displacement vector fields from image sequences**. *In Proceedings of IEEE Workshop on Motion: Representation and Analysis*, pages 101–6, 1986. 28

[50] A Singh. **Optic flow computation: A unified perspective**. *IEEE Computer Society Press*, 1992. 28

[51] M A Sutton, W J Walters, W H Peters, W F Ranson, and S R McNeil. **Determination of displacement using an improved digital correlation method**. *Image and Vision Computing*, **1(3)**:133–9, 1983. 28

[52] T Camus. **Real-time quantized optical flow**. *Real-Time Imaging*, **3(2)**:71–86, 1997. 28

[53] J Banks and P Corke. **Quantitative evaluation of matching methods and validity measures for stereo vision**. *International Journal of Robotics Research*, **20(7)**:512–32, 2001. 28

[54] B Galvin, B McCane, K Novins, D Mason, and S Mills. **Recovering motion fields: An evaluation of eight optical flow algorithms**. *In Proceedings of the Ninth British Machine Vision Conference*, pages 195–204, 1998. 28

[55] B K P. Horn and B G. Schunck. **Determining optical flow**. *Artificial Intelligence*, **13(1-3)**:185–203, 1981. 28, 49, 51, 52, 54

[56] H H. Nagel. **On the estimation of optical flow: relations between different approaches and some new results**. *Artificial Intelligence*, **33(3)**:299–324, 1987. 28

[57] M Bjorkman. **Real Time Motion and Stereo Cues for Active Visual Observers**. *PhD thesis, Numerical Analysis and Computing Science, Royal Institute of Technology, Sweden*, 2002. 28

[58] Denis N Lee, H Kalmus, DN Lee, and H Kalmus. **The optic flow field: The foundation of vision [and discussion]**. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, **290**(1038):169–179, 1980. 29, 74

[59] J. Barron, D. Fleet, and S. Beauchemin. **Performance of optical flow techniques.** *International Journal of Computer Vision*, **12**(1):43–77, 1994. 30

[60] B. D. Lucas and T. Kanade. **An iterative image registration technique with an application to stereo vision**. In *Imaging understanding workshop*, pages 121–120, 1981. 30

[61] Ramesh Jain. **Direct computation of the focus of expansion**. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (1):58–64, 1983. 36

[62] Shahriar Negahdaripour and Berthold KP Horn. **A direct method for locating the focus of expansion**. *Computer vision, graphics, and image processing*, **46**(3):303–326, 1989. 36

[63] S. Negahdaripour and V. Ganesan. **Simple direct computation of the FOE with confidence measures**. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pages 228–235, 1992. 36

[64] Roger Y Tsai and Thomas S Huang. **Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces**. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (1):13–27, 1984. 37

[65] H Christopher Longuet-Higgins and Kvetoslav Prazdny. **The interpretation of a moving retinal image**. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, **208**(1173):385–397, 1980. 37

[66] Anna R Bruss and Berthold KP Horn. **Passive navigation**. *Computer Vision, Graphics, and Image Processing*, **21**(1):3–20, 1983. 37

[67] José Rosa Kuiaski, André Eugênio Lazzaretti, and Hugo Vieira Neto. **Focus of Expansion Estimation for Motion Segmentation from a Single Camera**. In *Anais do VII Workshop de Visão Computacional (WVC 2011)*, pages 272–277, Curitiba, Brazil, 2011. 37

[68] Guillem Alenya, Amaury Nègre, and James L Crowley. **A comparison of three methods for measure of time to contact**. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4565–4570. IEEE, 2009. 39

[69] David N Lee et al. **A theory of visual control of braking based on information about time-to-collision**. *Perception*, **5**(4):437–459, 1976. 39

[70] Massimo Tistarelli, Giulio Sandini, and Giulio S. **On the Advantages of Polar and Log-polar Mapping for Direct Estimation of Time-to-impact from Optical Flow**. *IEEE Trans. on PAMI*, **15**:401–410, 1992. 39, 40

[71] B.K.P. Horn, Y. Fang, and I. Masaki. **Time to contact relative to a planar surface**. In *Intelligent Vehicles Symposium, 2007 IEEE*, pages 68–74. IEEE, 2007. 39

[72] Amaury Ngre, Christophe Braillon, James L. Crowley, and Christian Laugier. **Real-Time Time-to-Collision from Variation of Intrin-**

sic Scale. In Oussama Khatib, Vijay Kumar, and Daniela Rus, editors, *Experimental Robotics, The 10th International Symposium on Experimental Robotics [ISER 06, July 6-10, 2006, Rio de Janeiro, Brazil]*, **39** of *Springer Tracts in Advanced Robotics*, pages 75–84. Springer, 2006. 39, 71

[73] Roberto Cipolla and Andrew Blake. **Surface orientation and time to contact from image divergence and deformation**. In *Computer VisionECCV'92*, pages 187–202. Springer, 1992. 40

[74] Ronald W McLeod, Helen E Ross, et al. **Optic-flow and cognitive factors in time-to-collision estimates**. *Perception*, **12**(4):417–423, 1983. 40

[75] William Schiff and Mary Lou Detwiler. **Information used in judging impending collision**. *Perception*, **8**(6):647–658, 1979. 40

[76] Bruce D. Lucas and Takeo Kanade. **An Iterative Image Registration Technique with an Application to Stereo Vision (IJCAI)**. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679, April 1981. 49, 55

[77] Jianbo Shi and Tomasi. **Good features to track**. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94*, pages 593–600. IEEE Comput. Soc. Press, June 1994. 55

[78] Hongche Liu, Tsai-Hong Hong, Martin Herman, and Rama Chellappa. **Accuracy vs. Efficiency Trade-offs in Optical Flow Algorithms**. In *Proceedings of the 4th European Conference on Computer Vision-Volume II - Volume II*, ECCV '96, pages 174–183, London, UK, 1996. Springer-Verlag. 72

[79] Tompkin James. **Optical Flow, An introduction**. Technical report, EngD VEIV, 2008. 72

[80] Giacomo Indiveri, Elisabetta Chicca, and Rodney Douglas. **A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity**. *Neural Networks, IEEE Transactions on*, **17**(1):211–221, 2006. 77

[81] MM Khan, DR Lester, Luis A Plana, A Rast, X Jin, E Painkras, and Stephen B Furber. **SpiNNaker: mapping neural networks onto a massively-parallel chip multiprocessor**. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 2849–2856. IEEE, 2008. 77