Event-based Vision: A Survey

Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jörg Conradt, Kostas Daniilidis, Davide Scaramuzza

Abstract— Event cameras are bio-inspired sensors that work radically different from traditional cameras. Instead of capturing images at a fixed rate, they measure per-pixel brightness changes asynchronously. This results in a stream of events, which encode the time, location and sign of the brightness changes. Event cameras posses outstanding properties compared to traditional cameras: very high dynamic range ($140\,\mathrm{dB}$ vs. $60\,\mathrm{dB}$), high temporal resolution (in the order of $\mu\mathrm{s}$), low power consumption, and do not suffer from motion blur. Hence, event cameras have a large potential for robotics and computer vision in challenging scenarios for traditional cameras, such as high speed and high dynamic range. However, novel methods are required to process the unconventional output of these sensors in order to unlock their potential. This paper provides a comprehensive overview of the emerging field of event-based vision, with a focus on the applications and the algorithms developed to unlock the outstanding properties of event cameras. We present event cameras from their working principle, the actual sensors that are available and the tasks that they have been used for, from low-level vision (feature detection and tracking, optic flow, etc.) to high-level vision (reconstruction, segmentation, recognition). We also discuss the techniques developed to process events, including learning-based techniques, as well as specialized processors for these novel sensors, such as spiking neural networks. Additionally, we highlight the challenges that remain to be tackled and the opportunities that lie ahead in the search for a more efficient, bio-inspired way for machines to perceive and interact with the world.

Index Terms—Event Cameras, Bio-Inspired Vision, Asynchronous Sensor, Low Latency, High Dynamic Range, Low Power.

1 Introduction and Applications

"The *brain is imagination, and that was exciting to me; I wanted to build a chip that could imagine something*[1]." that is how Misha Mahowald, a graduate student at Caltech in 1986, started to work with Prof. Carver Mead on the stereo problem from a joint biological and engineering perspective. A couple of years later, in 1991, the image of a cat in the cover of Scientific American [1], acquired by a novel "Silicon Retina" mimicking the neural architecture of the eye, showed a new, powerful way of doing computations, igniting the emerging field of neuromorphic engineering. Today, we still pursue the same visionary challenge: understanding how the brain works and building one on a computer chip. Current efforts include flagship billion-dollar projects, such as the Human Brain Project and the Blue Brain Project in Europe, the U.S. BRAIN (Brain Research through Advancing Innovative Neurotechnologies) Initiative (presented by the U.S. President), and China's and Japan's Brain projects.

This paper provides an overview of the bio-inspired technology of silicon retinas, or "event cameras", such as [2], [3], [4], [5], with a focus on their application to solve classical as well as new computer vision and robotic tasks. Sight is, by far, the dominant sense in humans to perceive the world, and, together with the brain, learn new things. In recent years, this technology has attracted a lot of attention from both academia and industry. This is due to the availability of prototype event cameras and the advantages that these devices offer to tackle problems that are currently unfeasible with standard frame-based image sensors (that provide stroboscopic synchronous sequences of 2D pictures).

Event cameras are *asynchronous* sensors that pose a *paradigm shift* in the way visual information is acquired. This is because they sample light based on the scene dynamics, rather than on a clock that has no relation to the viewed scene. Their advantages are: very high temporal resolution and low latency (both in the order of microseconds), very high dynamic range ($140\,\mathrm{dB}$ vs. $60\,\mathrm{dB}$ of standard cameras), and low power consumption. Hence, event cameras have a large potential for robotics and wearable applications in challenging scenarios for standard cameras, such as high speed and high dynamic range. Although event cameras have become commercially available only since 2008 [2], the recent body of literature on these new sensors[2] as well as the recent plans for mass production claimed by companies, such as Samsung [5] and Prophesee[3], highlight that there is a big commercial interest in exploiting these novel vision sensors for mobile robotic, augmented and virtual reality (AR/VR), and video game applications. However, because event cameras work in a fundamentally different way from standard cameras, measuring per-pixel brightness changes (called "*events*") asynchronously rather than measuring "absolute" brightness at constant rate, novel methods are required to process their output and unlock their potential.

- *G. Gallego and D. Scaramuzza are with the Dept. of Informatics University of Zurich and Dept. of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland. Tobi Delbruck is with the Dept. of Information Technology and Electrical Engineering, ETH Zurich, at the Inst. of Neuroinformatics, University of Zurich and ETH Zurich, Zurich, Switzerland. Garrick Orchard is with Intel Corp., CA, USA. Chiara Bartolozzi is with the Italian institute of Technology, Genoa, Italy. Brian Taba is with IBM Research, CA, USA. Andrea Censi is with the Dept. of Mechanical and Process Engineering, ETH Zurich, Switzerland. Stefan Leutenegger and Andrew Davison are with Imperial College London, London, UK. Jörg Conradt is with KTH Royal Institute of Technology, Stockholm, Sweden. Kostas Daniilidis is with University of Pennsylvania, PA, USA. July 27, 2019*

1. https://youtu.be/FKemf6Idkd0?t=67

2. https://github.com/uzh-rpg/event-based_vision_resources
3. http://rpg.ifi.uzh.ch/ICRA17_event_vision_workshop.html

Applications of Event Cameras: Typical scenarios where event cameras offer advantages over other sensing modalities include real-time interaction systems, such as robotics or wearable electronics [6], where operation under uncontrolled lighting conditions, latency, and power are important [7]. Event cameras are used for object tracking [8], [9], [10], [11], [12], surveillance and monitoring [13], [14], object recognition [15], [16], [17], [18] and gesture control [19], [20]. They are also used for depth estimation [21], [22], [23], [24], [25], [26], 3D panoramic imaging [27], structured light 3D scanning [28], optical flow estimation [26], [29], [30], [31], [32], [33], [34], high dynamic range (HDR) image reconstruction [35], [36], [37], [38], mosaicing [39] and video compression [40]. In ego-motion estimation, event cameras have been used for pose tracking [41], [42], [43], and visual odometry and Simultaneous Localization and Mapping (SLAM) [44], [45], [46], [47], [48], [49], [50]. Event-based vision is a growing field of research, and other applications, such as image deblurring [51] or star tracking [52], are expected to appear as event cameras become widely available.

Outline: The rest of the paper is organized as follows. Section 2 presents event cameras, their working principle and advantages, and the challenges that they pose as novel vision sensors. Section 3 discusses several methodologies commonly used to extract information from the event camera output, and discusses the biological inspiration behind some of the approaches. Section 4 reviews applications of event cameras, from low-level to high-level vision tasks, and some of the algorithms that have been designed to unlock their potential. Opportunities for future research and open challenges on each topic are also pointed out. Section 5 presents neuromorphic processors and embedded systems. Section 6 reviews the software, datasets and simulators to work on event cameras, as well as additional sources of information. The paper ends with a discussion (Section 7) and conclusions (Section 8).

2 Principle of Operation of Event Cameras

In contrast to standard cameras, which acquire full images at a rate specified by an external clock (e.g., 30 fps), event cameras, such as the Dynamic Vision Sensor (**DVS**) [2], [53], [54], [55], [56], respond to *brightness changes* in the scene *asynchronously* and *independently* for every pixel (Fig. 1). Thus, the output of an event camera is a variable data-rate sequence of digital "events" or "spikes", with each event representing a change of brightness (log intensity)[4] of predefined magnitude at a pixel at a particular time[5] (Fig. 1, top right) (Section 2.4). This encoding is inspired by the spiking nature of biological visual pathways (Section 3.3). Each pixel memorizes the log intensity each time it sends an event, and continuously monitors for a change of sufficient

4. *Brightness* is a perceived quantity; for brevity we use it to refer to log intensity since they correspond closely for uniformly-lighted scenes.

5. Nomenclature: "Event cameras" output data-driven events that signal a place and time. This nomenclature has evolved over the past decade: originally they were known as address-event representation (AER) silicon retinas, and later they became event-based cameras. In general, events can signal any kind of information (intensity, local spatial contrast, etc.), but over the last five years or so, the term "event camera" has unfortunately become practically synonymous with the particular representation of brightness change output by DVS's.

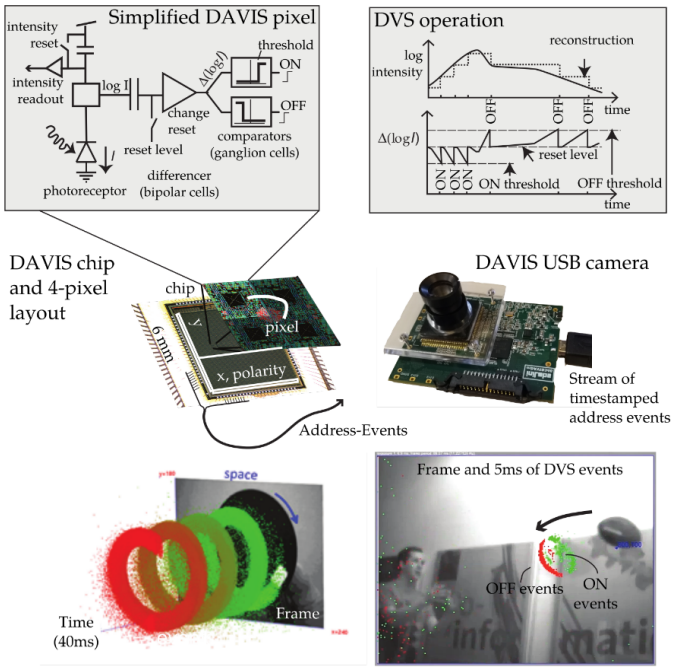


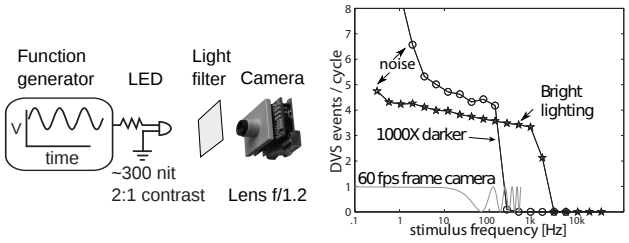
Figure 1. Summary of the DAVIS camera [4], [57], comprising an event-based dynamic vision sensor (DVS [56]) and a frame-based active pixel sensor (**APS**) in the same pixel array, sharing the same photodiode in each pixel. **Top left:** simplified circuit diagram of the DAVIS pixel. **Top right:** schematic of the operation of a DVS pixel, converting light into events. **Center:** pictures of the DAVIS chip and USB camera. **Bottom left:** space-time view, on the image plane, of frames and events caused by a spinning dot. **Bottom right:** frame and overlaid events of a natural scene; the frames lag behind the low-latency events. Images adapted from [4], [58].

magnitude from this memorized value (Fig. 1, top left). When the change exceeds a threshold, the camera sends an event, which is transmitted from the chip with the x, y location, the time t, and the 1-bit polarity p of the change (i.e., brightness increase ("ON") or decrease ("OFF")). This event output is illustrated in Fig. 1, bottom.

The events are transmitted from the pixel array to periphery and then out of the camera using a shared digital output bus, typically by using some variety of address-event representation (**AER**) readout [59], [60]. This AER bus can become saturated, which perturbs the times that events are sent. Event cameras achieve readout rates ranging from 2 MHz [2] to 300 MHz [5], depending on the chip and type of hardware interface.

Hence, event cameras are data-driven sensors: their output depends on the amount of motion or brightness change in the scene. The faster the motion, the more events per second are generated, since each pixel adapts its delta modulator sampling rate to the rate of change of the log intensity signal that it monitors. Events are timestamped with microsecond resolution and are transmitted with sub-millisecond latency, which make these sensors react quickly to visual stimuli.

The incident light at a pixel is a product of scene illumination and surface reflectance. Thus, a log intensity change in the scene generally signals a reflectance change (because usually the illumination is constant and the log of a product is the sum of the logs). Thus, these reflectance changes are mainly a result from movement of objects in the field of



(a) Measurement setup (b) Measured responses for two DC levels of illumination.

Figure 2. Shows the event transfer function from a single DVS pixel in response to sinusoidal LED stimulation. The background (BG) events cause additional ON events at very low frequencies. The 60 fps camera curve shows the transfer function including aliasing from frequencies above the Nyquist frequency. Adapted from [2].

view. That is why the DVS brightness change events have a built-in invariance to scene illumination [2].

Comparing Bandwidths of DVS Pixels and Frame-based Camera: Although DVS pixels are fast, like any physical transducer, they have a finite bandwidth: if the incoming light intensity varies too quickly, the front end photoreceptor circuits filter out the variations. The rise and fall time that is analogous to the exposure time in standard image sensors is the reciprocal of this bandwidth. Fig. 2 shows an example of measured DVS pixel frequency response. The measurement setup (Fig. 2a) uses a sinusoidally-varying generated signal to measure the response. Fig. 2b shows that, at low frequencies, the DVS pixel produces a certain number of events per cycle. Above some cutoff frequency, the variations are filtered out by the photoreceptor dynamics and the number of events per cycle drops. This cutoff frequency is a monotonically increasing function of light intensity. At the brighter light intensity, the DVS pixel bandwidth is about 3 kHz, equivalent to an exposure time of about 300 µs. At 1000× lower intensity, the DVS bandwidth is reduced to about 300 Hz. Even when the LED brightness is reduced by a factor of 1000, the frequency response of DVS pixels is ten times higher than the 30 Hz Nyquist frequency from a 60 fps image sensor. Also, the frame-based camera aliases frequencies above the Nyquist frequency back to the baseband, whereas the DVS pixel does not due to the continuous time response.

2.1 Event Camera Types

The first silicon retina was developed by Mahowald and Mead at Caltech during the period 1986-1992, in Ph.D. thesis work [61] that was awarded the prestigious Clauser prize[6]. Mahowald and Mead's sensor had logarithmic pixels, was modeled after the three-layer Kufler retina, and produced as output spike events using the AER protocol. However, it suffered from several shortcomings: each wire-wrapped retina board required precise adjustment of biasing potentiometers; there was considerable mismatch between the responses of different pixels; and pixels were too large to be a device of practical use. Over the next decade the neuromorphic community developed a series of silicon retinas. A summary of these developments is provided in [60]. The DVS event camera had its genesis in a frame-based silicon retina design where the continuous-time photoreceptor was capacitively coupled to a readout circuit that was reset each time the pixel was sampled [62]. More recent event camera technology has been reviewed in the electronics and neuroscience literature [6], [60], [63], [64], [65], [66].

Although surprisingly many applications can be solved by only processing events (i.e., brightness changes), it became clear that some also require some form of static output (i.e., "absolute" brightness). To address this shortcoming, there have been several developments of cameras that concurrently output dynamic and static information.

The Asynchronous Time Based Image Sensor (**ATIS**) [3], [67] has pixels that contain a DVS subpixel that triggers another subpixel to read out the absolute intensity. The trigger resets a capacitor to a high voltage. The charge is bled away from this capacitor by another photodiode. The brighter the light, the faster the capacitor discharges. The ATIS intensity readout transmits two more events coding the time between crossing two threshold voltages. This way, only pixels that change provide their new intensity values. The brighter the illumination, the shorter the time between these two events. The ATIS achieves large static dynamic range (>120 dB). However, the ATIS has the disadvantage that pixels are at least double the area of DVS pixels. Also, in dark scenes the time between the two intensity events can be long and the readout of intensity can be interrupted by new events ([68] proposes a workaround to this problem).

The widely-used Dynamic and Active Pixel Vision Sensor (**DAVIS**) illustrated in Fig. 1 combines a conventional active pixel sensor (**APS**) [69] in the same pixel with DVS [4], [57]. The advantage over ATIS is a much smaller pixel size since the photodiode is shared and the readout circuit only adds about 5 % to the DVS pixel area. Intensity (APS) frames can be triggered on demand, by analysis of DVS events, although this possibility is seldom exploited[7]. However, the APS readout has limited dynamic range (55 dB) and like a standard camera, it is redundant if the pixels do not change.

Commercial Cameras: These and other types or varieties of DVS-based event cameras are developed commercially by companies iniVation, Insightness, Samsung, CelePixel, and Prophesee; some of these companies offer development kits. Several developments are currently poised to enter mass production, with the limiting factor being pixel size; the most widely used event cameras have quite large pixels: DVS128 (40 µm), ATIS (30 µm), DAVIS240 and DAVIS346 (18.5 µm). The smallest published DVS pixel [5], by Samsung, is 9 µm; while conventional global shutter industrial APS are typically in the range of 2 µm–4 µm. Low spatial resolution is certainly a limitation for application, although many of the seminal publications are based on the 128 × 128 pixel DVS128 [56]. The DVS with largest published array size has only about VGA spatial resolution (768 × 640 pixels [70]).

2.2 Advantages of Event cameras

Event cameras present numerous advantages over standard cameras:

6. http://www.gradoffice.caltech.edu/current/clauser

7. https://github.com/SensorsINI/jaer/blob/master/src/eu/seebetter/ini/chips/davis/DavisAutoShooter.java

High Temporal Resolution: monitoring of brightness changes is fast, in analog circuitry, and the read-out of the events is digital, with a $1\,\mathrm{MHz}$ clock, which means that events are detected and timestamped with microsecond resolution. Therefore, event cameras can capture very fast motions, without suffering from motion blur typical of frame-based cameras.

Low Latency: each pixel works independently and there is no need to wait for a global exposure time of the frame: as soon as the change is detected, it is transmitted. Hence, event cameras have minimal latency: about $10\,\mu\mathrm{s}$ on the lab bench, and sub-millisecond in the real world.

Low Power: Because event cameras transmit only brightness changes, and thus remove redundant data, power is only used to process changing pixels. At the die level, most event cameras use on the order of $10\,\mathrm{mW}$, and there are prototypes that achieve less than $10\,\mu\mathrm{W}$. Embedded event-camera systems where the sensor is directly interfaced to a processor have demonstrated system-level power consumption (i.e., sensing plus processing) of $100\,\mathrm{mW}$ or less [19], [71], [72], [73], [74].

High Dynamic Range (HDR). The very high dynamic range of event cameras ($>120\,\mathrm{dB}$) notably exceeds the $60\,\mathrm{dB}$ of high-quality, frame-based cameras, making them able to acquire information from moonlight to daylight. It is due to the facts that the photoreceptors of the pixels operate in logarithmic scale and each pixel works independently, not waiting for a global shutter. Like biological retinas, DVS pixels can adapt to very dark as well as very bright stimuli.

2.3 Challenges Due To The Novel Sensing Paradigm

Event cameras represent a paradigm shift in acquisition of visual information. Hence, they pose some challenges:

1) *Novel Algorithms:* The output of event cameras is fundamentally different from that of standard cameras. Thus, frame-based vision algorithms designed for image sequences are not directly applicable. Specifically, events depend not only on the scene brightness, but also on the current and past motion between the scene and the camera. Novel algorithms are thus required to process the event camera output to unlock the advantages of the sensor.

2) *Information Processing:* Each event contains binary (increase/decrease) brightness change information, as opposed to the grayscale information that standard cameras provide. Thus, it poses the question: what is the best way to extract information from the events relevant for a given task?

3) *Noise and Dynamic Effects:* All vision sensors are noisy because of the inherent shot noise in photons and from transistor circuit noise, and they also have non-idealities. This situation is especially true for event cameras, where the process of quantization of brightness change information is complex and has not been completely characterized. Hence, how can noise and non-ideal effects be modeled to better extract meaningful information from the events?

2.4 Event Generation Model

An event camera [2] has independent pixels that respond to changes in their log photocurrent $L \doteq \log(I)$ ("brightness"). Specifically, in a noise-free scenario, an event $e_k \doteq (\mathbf{x}_k, t_k, p_k)$ is triggered at pixel $\mathbf{x}_k \doteq (x_k, y_k)^\top$ and at time t_k as soon as the brightness increment since the last event at the pixel, i.e.

$$\Delta L(\mathbf{x}_k, t_k) \doteq L(\mathbf{x}_k, t_k) - L(\mathbf{x}_k, t_k - \Delta t_k), \tag{1}$$

reaches a temporal contrast threshold $\pm C$ (with $C > 0$) (Fig. 1 top right), i.e.,

$$\Delta L(\mathbf{x}_k, t_k) = p_k\, C, \tag{2}$$

where Δt_k is the time elapsed since the last event at the same pixel, and the polarity $p_k \in \{+1, -1\}$ is the sign of the brightness change [2].

The contrast sensitivity C is determined by the pixel bias currents [75], [76], which set the speed and threshold voltages of the change detector in Fig. 1 and are generated by an on-chip digitally-programmed bias generator. The sensitivity C can be estimated knowing these currents [75]. In practice, positive ("ON") and negative ("OFF") events may be triggered according to different thresholds, C^+, C^-. Typical DVS's can set thresholds between $15\,\%$–$50\,\%$ illumination change. The lower limit on C is determined by noise and pixel to pixel mismatch (variability) of C; setting C too low results in a storm of noise events, starting from pixels with low values of C. Experimental DVS's with higher photoreceptor gain are capable of lower thresholds, e.g., $1\,\%$ [77], [78], [79]; however these values are only obtained under very bright illumination and ideal conditions. Fundamentally, the pixel must react to a small change in the photocurrent in spite of the shot noise present in this current. This shot noise limitation sets the relation between threshold and speed of the DVS under a particular illumination and desired detection reliability condition [79], [80].

Events and the Temporal Derivative of Brightness: Eq. (2) states that event camera pixels set a threshold on magnitude of the brightness change since the last event happened. For a small Δt_k, such an increment (2) can be approximated using Taylor's expansion by $\Delta L(\mathbf{x}_k, t_k) \approx \frac{\partial L}{\partial t}(\mathbf{x}_k, t_k)\Delta t_k$, which allows us to interpret the events as providing information about the temporal derivative of brightness:

$$\frac{\partial L}{\partial t}(\mathbf{x}_k, t_k) \approx \frac{p_k\, C}{\Delta t_k}. \tag{3}$$

This is an indirect way of measuring brightness, since with standard cameras we are used to measuring absolute brightness. This interpretation may be taken into account to design principled event-based algorithms, such as [37], [81].

Events are Caused by Moving Edges: Assuming constant illumination, linearizing (2) and using the constant brightness assumption one can show that events are caused by moving edges. For small Δt, the intensity increment (2) can be approximated by[8]:

$$\Delta L \approx -\nabla L \cdot \mathbf{v} \Delta t, \tag{4}$$

that is, it is caused by an brightness gradient $\nabla L(\mathbf{x}_k, t_k) = (\partial_x L, \partial_y L)^\top$ moving with velocity $\mathbf{v}(\mathbf{x}_k, t_k)$ on the image plane, over a displacement $\Delta \mathbf{x} \doteq \mathbf{v}\Delta t$. As the dot product (4) conveys: (*i*) if the motion is parallel to the edge, no

8. Eq. (4) can be shown [82] by substituting the brightness constancy assumption (i.e., optical flow constraint) $\frac{\partial L}{\partial t}(\mathbf{x}(t), t) + \nabla L(\mathbf{x}(t), t) \cdot \dot{\mathbf{x}}(t) = 0$, with image-point velocity $\mathbf{v} \equiv \dot{\mathbf{x}}$, in Taylor's approximation $\Delta L(\mathbf{x}, t) \doteq L(\mathbf{x}, t) - L(\mathbf{x}, t - \Delta t) \approx \frac{\partial L}{\partial t}(\mathbf{x}, t)\Delta t$.

event is generated since $\mathbf{v} \cdot \nabla L = 0$; (*ii*) if the motion is perpendicular to the edge ($\mathbf{v} \parallel \nabla L$) events are generated at the highest rate (i.e., minimal time is required to achieve a brightness change of size $|C|$).

Probabilistic Event Generation Models: Equation (2) is an idealized model for the generation of events. A more realistic model takes into account sensor noise and transistor mismatch, yielding a mixture of frozen and temporally varying stochastic triggering conditions represented by a probability function, which is itself a complex function of local illumination level and sensor operating parameters. The measurement of such probability density was shown in [2] (for the DVS), suggesting a normal distribution centered at the contrast threshold C. The 1-σ width of the distribution is typically 2-4% temporal contrast. This event generation model can be included in emulators [83] and simulators [84] of event cameras, and in estimation frameworks to process the events, as demonstrated in [39], [82]. Other probabilistic event generation models have been proposed, such as: the likelihood of event generation being proportional to the magnitude of the image gradient [45] (for scenes where large intensity gradients are the source of most event data), or the likelihood being modeled by a mixture distribution to be robust to sensor noise [43]. Future even more realistic models will include the refractory period after each event (during which the pixel is blind to change), and bus congestion [85].

The above event generation models are simple, developed to some extent based on sensor noise characterization. Just like standard image sensors, DVS's also have fixed pattern noise (**FPN**[9]), but in DVS it manifests as pixel-to-pixel variation in the event threshold. Standard DVS's can achieve minimum $C \approx \pm 15\,\%$, with a standard deviation of about $2.5\,\%$–$4\,\%$ contrast between pixels [2], [86], and there have been attempts to measure pixelwise thresholds by comparing brightness changes due to DVS events and due to differences of consecutive DAVIS APS frames [40]. However, understanding of *temporal* DVS pixel and readout noise is preliminary [2], [78], [85], [87], and noise filtering methods have been developed mainly based on computational efficiency, assuming that events from real objects should be more correlated spatially and temporally than noise events [60], [88], [89], [90], [91]. We are far from having a model that can predict event camera noise statistics under arbitrary illumination and biasing conditions. Solving this challenge would lead to better estimation methods.

2.5 Event Camera Availability

Table 1 shows currently popular event cameras. Some of them also provide absolute intensity (e.g., grayscale) output, and some also have an integrated Inertial Measurement Unit (**IMU**) [93]. IMUs act as a vestibular sense that is valuable for improving camera pose estimation, such as in visual-inertial odometry (Section 4.5).

Cost: Currently, a practical obstacle to adoption of event camera technology is the high cost of several thousand dollars per camera, similar to the situation with early time of flight, structured lighting and thermal cameras. The high costs are due to non-recurring engineering costs for the silicon design and fabrication (even when much of it is provided by research funding) and the limited samples available from prototype runs. It is anticipated that this price will drop precipitously once this technology enters mass production.

Pixel Size: Since the first practical event camera [2] there has been a trend mainly to increase resolution, increase readout speed, and add features, such as: gray level output (e.g., as in ATIS and DAVIS), integration with IMU [93] and multi-camera event timestamp synchronization [94]. Only recently has the focus turned more towards the difficult task of reducing pixel size for economical mass production of sensors with large pixel arrays. From the $40\,\mu\mathrm{m}$ pixels of the 128×128 DVS in $350\,\mathrm{nm}$ technology in [2], the smallest published pixel has shrunk to $9\,\mu\mathrm{m}$ in $90\,\mathrm{nm}$ technology in the 640×480 pixel DVS in [5]. Event camera pixel size has shrunk pretty closely following feature size scaling, which is remarkable considering that a DVS pixel is a mixed-signal circuit, which generally do not scale following technology. However, achieving even smaller pixels will be difficult and may require abandoning the strictly asynchronous circuit design philosophy that the cameras started with. Camera cost is constrained by die size (since silicon costs about \$5-\$10/cm^2 in mass production), and optics (designing new mass production miniaturized optics to fit a different sensor format can cost tens of millions of dollars).

Fill Factor: A major obstacle for early event camera mass production prospects was the limited fill factor of the pixels (i.e, the ratio of a pixel's light sensitive area to its total area). Because the pixel circuit is complex, a smaller pixel area can be used for the photodiode that collects light. For example, a pixel with $20\,\%$ fill factor throws away 4 out of 5 photons. Obviously this is not acceptable for optimum performance; nonetheless, even the earliest event cameras could sense high contrast features under moonlight illumination [2]. Early CMOS image sensors (**CIS**) dealt with this problem by including microlenses that focused the light onto the pixel photodiode. What is probably better, however, is to use back-side illumination technology (**BSI**). BSI flips the chip so that it is illuminated from the back, so that in principle the entire pixel area can collect photons. Nearly all smartphone cameras are now back illuminated, but the additional cost and availability of BSI fabrication has meant that only recently BSI event cameras were first demonstrated [5], [95]. BSI also brings problems: light can create additional 'parasitic' photocurrents that lead to spurious 'leak' events [75].

Advanced Event Cameras

There are active developments of more advanced event cameras that are not available commercially, although many can be used in scientific collaborations with the developers. This section discusses issues related to advanced camera developments and the types of new cameras that are being developed.

Color: Most diurnal animals have some form of color vision, and most conventional cameras offer color sensitivity. Early attempts at color sensitive event cameras [96], [97], [98] tried to use the "vertacolor" principle of splitting colors according to the amount of penetration of the different light wavelengths into silicon, pioneered by Foveon [99],

9. https://en.wikipedia.org/wiki/Fixed-pattern_noise

Table 1
Comparison between different commercialized event cameras.

	DVS128 [2]	DAVIS240 [4]	DAVIS346	ATIS [3]	Gen3 CD [92]	Gen3 ATIS [92]	DVS-Gen2 [5]	CeleX-IV [70]
Supplier	iniVation	iniVation	iniVation	Prophesee	Prophesee	Prophesee	Samsung	CelePixel
Year	2008	2014	2017	2011	2017	2017	2017	2017
Resolution (pixels)	128×128	240×180	346×260	304×240	640×480	480×360	640×480	768×640
Latency (µs)	12μs @ 1klux	12μs @ 1klux	20	3	40 - 200	40 - 200	65 - 410	-
Dynamic range (dB)	120	120	120	143	> 120	> 120	90	100
Min. contrast sensitivity (%)	17	11	14.3 - 22.5	13	12	12	9	-
Die power consumption (mW)	23	5 - 14	10 - 170	50 - 175	36 - 95	36 - 95	27 - 50	-
Camera Max. Bandwidth (Meps)	1	12	12	-	66	66	300	200
Chip size (mm^2)	6.3×6	5×5	8×6	9.9×8.2	9.6×7.2	9.6×7.2	8×5.8	-
Pixel size (μm^2)	40×40	18.5×18.5	18.5×18.5	30×30	15×15	20×20	9×9	18×18
Fill factor (%)	8.1	22	22	20	25	25	100	9
Supply voltage (V)	3.3	1.8 & 3.3	1.8 & 3.3	1.8 & 3.3	1.8	1.8	1.2 & 2.8	3.3
Stationary noise (ev /pix /s) at 25C	0.05	0.1	0.1	NA	0.1	0.1	0.03	-
CMOS technology (µm)	0.35 2P4M	0.18 1P6M MIM	0.18 1P6M MIM	0.18 1P6M	0.18 1P6M CIS	0.18 1P6M CIS	0.09 1P5M BSI	0.18 1P6M CIS
Grayscale output	no	yes	yes	yes	no	yes	no	yes
Grayscale dynamic range (dB)	NA	55	56.7	130	NA	> 100	NA	-
Max. framerate (fps)	NA	35	40	NA	NA	NA	NA	-
IMU output	no	1 kHz	1 kHz	no	1 kHz	1 kHz	no	no

[100]. However, it resulted in poor color separation performance. So far, there are few publications of practical color event cameras, with either integrated color filter arrays (**CFA**) [101], [102], [103] or color-splitter prisms [104]; splitters have a much higher cost than CFA.

Higher Contrast Sensitivity: Efforts have been made to improve the temporal contrast sensitivity of event cameras (see Section 2.4), leading to experimental sensors with higher sensitivity [77], [78], [79] (down to laboratory condition $\sim 1\,\%$). These sensors are based on variations of the idea of a thermal bolometer [105], i.e., increasing the gain before the change detector (Fig. 1) to reduce the input-referred FPN. However this intermediate preamplifier requires active gain control to avoid clipping. Increasing the contrast sensitivity is possible, at the expense of decreasing the dynamic range (e.g., [5]).

3 Event Processing Paradigms

One of the key questions of the paradigm shift posed by event cameras is how to extract meaningful information from the event stream to fulfill a given task. This is a very broad question, since the answer is application dependent, and it drives the algorithmic design of the task solver.

Depending on how many events are processed simultaneously, two categories of algorithms can be distinguished: *1)* methods that operate on an *event-by-event basis*, where the state of the system (the estimated unknowns) can change upon the arrival of a single event, and *2)* methods that operate on *groups of events*. Using a temporal sliding window, methods based on groups of events can provide a state update upon the arrival of a single event (sliding by one event). Hence, the distinction between both categories is deeper: an event alone does not provide enough information for estimation, and so additional information, in the form of events or extra knowledge, is needed. The above categorization refers to this implicit source of additional information.

Orthogonally, depending on how events are processed, we can distinguish between model-based approaches and model-free (i.e., data-driven, machine learning) approaches. Assuming events are processed in an optimization framework, another classification concerns the type of objective or loss function used: geometric- vs. photometric-based (e.g., a function of the event polarity or the event activity).

Each category presents methods with advantages and disadvantages and current research focuses on exploring the possibilities that each method can offer.

3.1 Event-by-Event

Model based: Event-by-event–based methods have been used for multiple tasks, such as feature tracking [46], pose tracking in SLAM systems [39], [43], [44], [45], [47], and image reconstruction [36], [37]. These methods rely on the availability of additional information (typically "appearance" information, such as grayscale images or a map of the scene), which may be provided by past events or by additional sensors. Then, each incoming event is compared against such information and the resulting mismatch provides innovation to update the system state. Probabilistic filters are the dominant framework of these type of methods because they naturally (*i*) handle asynchronous data, thus providing minimum processing latency, preserving the sensor's characteristics, and (*ii*) aggregate information from multiple small sources (e.g., events).

Model free: Model free event-by-event algorithms typically take the form of a multi-layer neural network (whether spiking or not) containing many parameters which must be derived from the event data. Networks trained with unsupervised learning typically act as feature extractors for a classifier (e.g. SVM), which still requires some labeled data for training [16], [17], [106]. If enough labeled data is available, supervised learning methods such as backpropagation can be used to train a network without the need for a separate classifier. Many approaches use *groups of events* during training (deep learning on frames), and later convert the trained network to a Spiking Neural Network (**SNN**) that processes data *event-by-event* [107], [108], [109], [110], [111]. Event-by-event model free methods have mostly been applied to classify objects [16], [17], [107], [108] or actions [19],

[20], [112], and have targeted embedded applications [107], often using custom SNN hardware [16], [19].

3.2 Groups of Events

Model based: Methods that operate on groups of events aggregate the information contained in the events to estimate the problem unknowns, usually without relying on additional data. Since each event carries little information and is subject to noise, several events must be processed together to yield a sufficient signal-to-noise ratio for the problem considered. This category can be further subdivided into two: (*i*) methods that *quantize temporal information* of the events and accumulate them into frames, possibly guided by application or computing power, to re-utilize traditional, image-based computer vision algorithms [113], [114], [115], and (*ii*) methods that *exploit the fine temporal information* of individual events for estimation, and therefore tend to depart from traditional computer vision algorithms [23], [26], [32], [49], [116], [117], [118], [119], [120], [121]. The review [7] quantitatively compares accuracy and computational cost for frame-based versus event-driven optical flow.

Events are processed differently depending on their *representation*. Some approaches use techniques for point sets [30], [46], [117], [122], reasoning in terms of geometric processing of the space-time coordinates of the events. Other methods process events as tensors: time-surfaces (pixel-map of last event timestamps) [88], [123], [124], event histograms [32], etc. Others, [23], [26], combine both: warping events as point sets to compute tensors for further analysis.

Model free (Deep Learning): So-called *model free* methods operating on groups of events typically consist of a deep neural network. Sample applications include classification [125], [126], steering angle prediction [127], [128], and estimation of optical flow [33], [129], [130], depth [129] or ego-motion [130]. These methods differentiate themselves mainly in the representation of the input (events) and in the loss functions that are optimized during training.

Since classical deep learning pipelines use tensors as inputs, events have to be converted into such a dense, multichannel *representation*. Several representations have been used, such as: pixelwise histograms of events [128], [131], maps of most recent timestamps [33], [129], [132] ("time surfaces" [17], [88]), or an interpolated voxel grid [38], [130], which better preserves the spatio-temporal nature of the events within a time interval. A general framework to convert event streams into grid-based representations is given in [133]. Alternatively, point set representations, which do not require conversion, have been recently explored [134], inspired by [135].

While *loss functions* in classification tasks use manually annotated labels, networks for regression tasks from events may be supervised by a third party ground truth (e.g., a pose) [128], [131] or by an associated grayscale image [33] to measure photoconsistency, or be completely unsupervised (depending only on the training input events) [129], [130]. Loss functions for unsupervised learning from events are studied in [121]. In terms of *architecture*, most networks have an encoder-decoder structure, as in Fig. 3. Such a structure allows the use of convolutions only, thus minimizing the number of network weights. Moreover, a loss function can be applied at every spatial scale of the decoder.

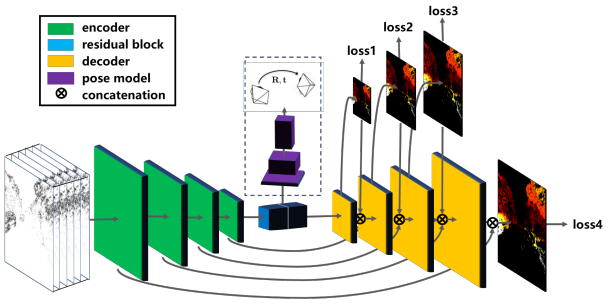


Figure 3. Network architecture for both, optical flow and ego-motion–depth networks. In the optical flow network, only the encoder-decoder section is used, while in the ego-motion and depth network, the encoder-decoder is used to predict depth, and the pose model predicts ego-motion. At training time, the loss is applied at each stage of the decoder, before being concatenated into the next stage of the network [130].

3.3 Biologically Inspired Visual Processing

The DVS [2] was inspired by the function of biological visual pathways, which have "transient" pathways dedicated to processing dynamic visual information in the so-called "where" pathway. Animals ranging from insects to humans all have these transient pathways. In humans, the transient pathway occupies about 30 % of the visual system. It starts with transient ganglion cells, which are mostly found in retina outside the fovea. It continues with magno layers of the thalamus and particular sublayers of area V1. It then continues to area MT and MST, which are part of the dorsal pathway where many motion selective cells are found [63]. The DVS corresponds to the part of the transient pathway(s) up to retinal ganglion cells.

Spiking Neural Network (SNN): Biological perception principles and computational primitives drive not only the design of event camera pixels but also some of the algorithms used to *process* the events. Artificial neurons, such as Leaky-Integrate and Fire or Adaptive Exponential, are computational primitives inspired in neurons found in the mammalian's visual cortex. They are the basic building blocks of artificial SNNs. A neuron receives input spikes ("events") from a small region of the visual space (a receptive field), which modify its internal state (membrane potential) and produce an output spike (action potential) when the state surpasses a threshold. Neurons are connected in a hierarchical way, forming an SNN. Spikes may be produced by pixels of the event camera or by neurons of the SNN. Information travels along the hierarchy, from the event camera pixels to the first layers of the SNN and then through to higher (deeper) layers. Most first layer receptive fields are based on Difference of Gaussians (selective to center-surround contrast), Gabor filters (selective to oriented edges), and their combinations. The receptive fields become increasingly more complex as information travels deeper into the network. In artificial neural networks, the computation performed by inner layers is approximated as a convolution. One common approach in artificial SNNs is to assume that a neuron will not generate any output spikes if it has not received any input spikes from the preceding SNN layer. This assumption allows computation to be skipped for such neurons. The result of this visual processing is almost simultaneous with the stimulus presentation [136], which

is very different from traditional convolutional networks, where convolution is computed simultaneously at all locations at fixed time intervals.

Tasks: Bio-inspired models have been adopted for several low-level visual tasks. For example, event-based *optical flow* can be estimated by using spatio-temporally oriented filters [88], [137], [138] that mimic the working principle of receptive fields in the primary visual cortex [139], [140]. The same type of oriented filters have been used to implement a spike-based model of *selective attention* [141] based on the biological proposal from [142]. Bio-inspired models from binocular vision, such as recurrent lateral connectivity and excitatory-inhibitory neural connections [143], have been used to solve the event-based *stereo* correspondence problem [61], [144], [145], [146], [147] or to control binocular vergence on humanoid robots [148]. The visual cortex has also inspired the hierarchical feature extraction model proposed in [149], which has been implemented in SNNs and used for *object recognition*. The performance of such networks improves the better they extract information from the precise timing of the spikes [150]. Early networks were hand-crafted (e.g., using Gabor filters) [71], but recent efforts let the network build receptive fields through brain-inspired learning, such as Spike-Timing Dependent Plasticity, yielding better recognition networks [106]. This research is complemented by approaches where more computationally inspired types of supervised learning, such as back-propagation, are used in deep networks to efficiently implement spiking deep convolutional networks [151], [152], [153], [154], [155].

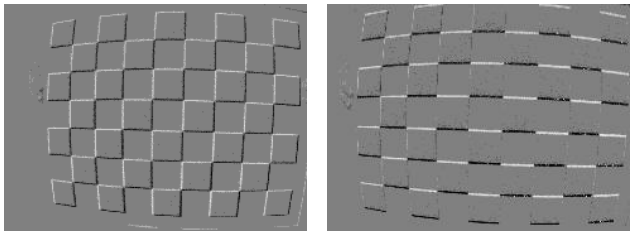
The advantages of the above methods over their traditional vision counterparts are lower latency and higher computational efficiency. To build small, efficient and reactive computational systems, *insect vision* is also a source of inspiration for event-based processing. To this end, systems for fast and efficient obstacle avoidance and target acquisition in small robots have been developed [156], [157], [158] based on models of neurons driven by DVS output that respond to looming objects and trigger escape reflexes.

4 Algorithms / Applications

In this section, we review several works on event-based vision, grouped according to the task addressed. We start with low-level vision on the image plane, such as feature detection, tracking, and optical flow estimation. Then, we discuss tasks that pertain to the 3D structure of the scene, such as depth estimation, structure from motion (SFM), visual odometry (VO), sensor fusion (visual-inertial odometry) and related subjects, such as intensity image reconstruction. Finally, we consider segmentation, recognition and coupling perception with control.

4.1 Feature Detection and Tracking

Feature detection and tracking on the image plane are fundamental building blocks of many vision tasks such as visual odometry, object segmentation and scene understanding. Event cameras enable tracking asynchronously, adapted to the dynamics of the scene and with low latency, high dynamic range and low power. Thus, they allow to track in the "blind" time between the frames of a standard camera.



(a) (b)

Figure 4. The challenge of data association. Panels (a) and (b) show events from the scene (a checkerboard) under two different motion directions: (a) diagonal and (b) up-down. These are intensity increment images obtained by accumulating events over a short time interval: pixels that do not change intensity are represented in gray, whereas pixels that increased or decreased intensity are represented in bright and dark, respectively. Clearly, it is not easy to establish event correspondences between (a) and (b) due to the changing appearance of the edge patterns with respect to the motion. Image adapted from [160].

Feature detection and tracking methods are typically application dependent. According to the scenario, we distinguish between methods designed for static cameras and methods designed for moving cameras. Since event cameras respond to the apparent motion of edge patterns in the scene, in the first scenario events are mainly caused by moving objects, whereas in the second scenario events are due to both, moving objects of interest ("foreground") as well as the moving background (due to the camera motion). Some fundamental questions driving the algorithmic design are: "what to detect/track?", "how to represent it using events?", "how to actually detect/track it?", and "what kind of distortions can be handled?". For example, objects of interest are usually represented by parametric models in terms of shape primitives (i.e., geometry-based) or edge patterns (i.e., appearance-based). The tracking strategy refers to how the transformation parameters of the model are updated upon the arrival of events. The model may be able to handle isometries, occlusions and other distortions of the object.

Challenges: Two main challenges of feature detection and tracking with event cameras are (*i*) overcoming the change of scene appearance conveyed by the events (Fig. 4), and (*ii*) dealing with sensor noise and non-linearities (neuromorphic sensors are known to be noisy [159]). Tracking requires the establishment of correspondences between events at different times (i.e., data association), which is difficult due to the above-mentioned varying appearance (Fig. 4). The problem simplifies if the absolute intensity of the pattern to be tracked (i.e., a time-invariant representation or "map" of the feature) is available. This may be provided by a standard camera colocated with the event camera or by image reconstruction techniques (Section 4.6).

Literature Review: Early event-based feature methods were very *simple* and focused on demonstrating the low-latency and low-processing requirements of event-driven vision systems, hence they assumed a static camera scenario and tracked moving objects as clustered blob-like sources of events [8], [9], [10], [13], [14], [161], circles [162] or lines [72]. They were used in traffic monitoring and surveillance [13], [14], [161], high-speed robotic target tracking [8], [10] and particle tracking in fluids [9] or microrobotics [162].

Tracking *complex*, high-contrast user-defined shapes has been demonstrated using event-by-event adaptations of the

Iterative Closest Point (ICP) algorithm [163], gradient descent [122], Monte-Carlo methods [164], or particle filtering [12]. The iterative methods in [122], [163] used a nearest-neighbor strategy to associate incoming events to the target shape and update its transformation parameters, showing very high-speed tracking (200 kHz equivalent frame rate). Complex objects, such as faces or human bodies, have been tracked with event cameras using part-based shape models [165], where objects are represented as a set of basic elements linked by springs [166]. The part trackers simply follow incoming blobs of events generated by ellipse-like shapes, and the elastic energy of this virtual mechanical system provides a quality criterion for tracking.

To some extent, all previous methods require *a priori* knowledge or user input to determine the objects to track. This restriction is valid for scenarios like tracking cars on a highway, microfluidic cells, or balls approaching a goal, since knowing the objects greatly simplifies the computations. But when the space of objects becomes larger, other methods determine distinctive, *natural features* to track by analyzing the events. As shown in [46], [167], such features are local patches of intensity gradients (edge patterns). Extending [46], features are built in [117] from motion-compensated events, producing point-set–based templates to which new events are registered. These features allowed to tackle the moving camera scenario in natural scenes [46], [120]. Also in this scenario, [49] proposed to apply traditional feature detectors [168] and trackers [169] on patches of motion-compensated event images [116]. Hence, motion-compensated events provide a useful representation of edge patterns, albeit it is subject to the apparent motion, and, therefore, suffers from tracking drift as event appearance changes over time. To remove drift, events can be combined with absolute intensity images, provided by a standard camera or built from past events (Section 4.6).

Combining Events and Frames: There is a growing body of literature leveraging the strengths of a combined frame- and event-based sensor (e.g., a DAVIS [4]). [46], [160], [167] present algorithms to automatically detect arbitrary edge patterns on the frames and track them asynchronously using events. Thus, they allow to track natural patterns in the scene and use them, e.g., for visual odometry [46].

Corner Detection and Tracking: Some works do not detect and track shapes but rather lower-level primitives, such as keypoints or "corners", directly on the event stream. Such primitives identify pixels of interest around which local features can be extracted without suffering from the aperture problem. The method in [170] computes corners as the intersection of two moving edges, which are obtained by fitting planes in the space-time stream of events. Plane fitting has also been used to estimate visual flow [30] and "event lifetime" [171]. Recently, extensions of popular frame-based keypoint detectors, such as Harris [168] and FAST [172], have been developed for event cameras [123], [173], by directly operating on events. Learning-based methods have also emerged [174]. The method in [124] additionally proposes a strategy to track event corners. Event corners find multiple applications, such as visual odometry or ego-motion segmentation [175].

Opportunities: In spite of the abundance of detection and tracking methods, they are rarely evaluated on common datasets for performance comparison. Establishing benchmark datasets [176] and evaluation procedures will foster progress in this topic. Also, in most algorithms, parameters are defined experimentally according to the tracking target. It would be desirable to have adaptive parameter tuning to increase the range of operation of the trackers. Learning-based feature detection and tracking methods also offer considerable room for research.

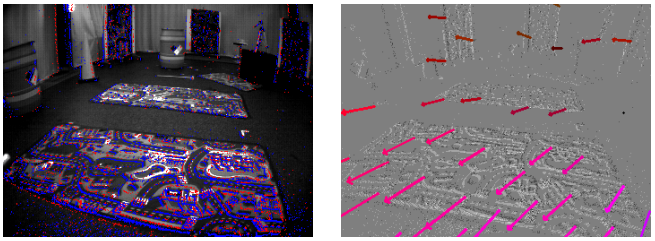
Table 2
Classification of optical flow methods according to whether they provide normal (N) or full flow (F), sparse (S) or dense (D) estimates, and whether they are model-based or model-free (Neural Network - NN), and neuro-biologically inspired or not.

Reference	N/F?	S/D?	Model?	Bio?
Delbruck [31], [88]	Normal	Sparse	Model	Yes
Benosman et al. [29], [31]	Full	Sparse	Model	No
Orchard et al. [137]	Full	Sparse	NN	Yes
Benosman et al. [30], [31]	Normal	Sparse	Model	No
Tschechne et al. [138]	Normal	Sparse	Model	Yes
Barranco et al. [177]	Normal	Sparse	Model	No
Barranco et al. [178]	Normal	Sparse	Model	No
Conradt et al. [73]	Normal	Sparse	Model	No
Brosch et al. [179]	Normal	Sparse	Model	Yes
Bardow et al. [32]	Normal	Dense	Model	No
Liu et al. [115], [180]	Normal	Sparse	Model	No
Gallego [26], Stoffregen [181]	Full	Sparse	Model	No
Haessig et al. [182]	Normal	Sparse	NN	Yes
Zhu et al. [33], [130]	Full	Dense	NN	No
Ye et al. [129], [132]	Full	Dense	NN	No
Paredes-Vallés [34]	Full	Sparse	NN	Yes

4.2 Optical Flow Estimation

Optical flow estimation refers to the problem of computing the velocity of objects on the image plane in its most general form, without prior knowledge about the scene geometry or camera motion. In static scenes, optical flow is due to the camera motion and the scene depth, and once the SLAM problem is solved (Section 4.4), flow can be trivially computed from the solution, as the so-called motion field. Next, we discuss the above-mentioned most general form of the problem, which is ill-posed and thus requires regularization with priors of different form. Event-based optical flow estimation is a challenging problem because of the unfamiliar way in which events encode visual information (Section 2.3): events provide neither absolute brightness nor spatially continuous data to exploit with neighboring pixels (since events are asynchronous). However, computing flow from events is attractive because of the fine timing information from the events that allows measuring high speed flow at low computational cost [7]. Event-based optical flow methods can be categorized according to different criteria (Table 2), such as: normal flow vs. full flow estimation, sparse vs. dense output, model-based vs. model-free, and whether they are bio-inspired or not. Let us present the methods using these criteria.

Normal vs. Full Flow: Early works, such as [29], [30], [88], estimated normal optical flow: the component of the normal flow perpendicular to the edge (i.e., parallel to the brightness gradient). Normal flow is moderately straightforward to compute using the pixel map of last timestamps of the events, also called the surface of active events [30], [88] (**SAE**). More recent methods, such as [26], [32], [33],



(a) Scene *Indoor Flying* from [183] (b) Sparse optical flow

Figure 5. Example of optical flow estimation with an event camera. (a) Events (with polarity in red/blue) overlaid on a grayscale frame; (b) Sparse optical flow computed using the Lucas-Kanade method [169] on event-accumulated frames, colored according to flow magnitude and direction; overlaid on events (positive: bright, and negative: dark).

[180], estimate the full optical flow (i.e., both tangential and normal components). Full optical flow is more informative than normal flow, but it is considerably harder to compute.

Sparse vs. Dense Flow: Optical flow methods can also be classified according to the type of output produced: (*i*) a sparse flow field (i.e., optical flow at a few pixels), such as [26], [29], [30], [31], [34], [178], [180], or (*ii*) a dense flow field (i.e., optical flow at every pixel), such as [32], [33], [129], [130], [132]. Event cameras respond to moving edges, and so they naturally provide information about flow at edges. These are the locations where the estimated optical flow is most reliable. Flow vectors computed at regions with no events (i.e., constant brightness regions) are due to interpolation or regularization, thus they are less reliable than those computed at edges.

Model-based vs. Model-free: Additionally, event-based optical flow methods can be categorized according to their design principle: model-based or model-free (i.e., data-based). Model-based approaches are grounded on several principles, such as time of flight of oriented edges [88], event-based adaptation of Lukas-Kanade optical flow [29], [31], local plane-fitting in space-time [30], phased-based methods [178], variational optimization [32], block-matching of increment-brightness images [115], [180] or contrast maximization [26]. In contrast, model-free methods are based on the availability of large amounts of event data paired with a neural network [33], [34], [129], [137], [182].

Computational Effort: Not all methods demand the same computational resources. Some methods [32], [33], [34], [129] require a GPU, and so they are computationally expensive compared to other, more lightweight, but possibly not as accurate, approaches [115]. There is an accuracy vs. efficiency trade off that has not been properly quantified yet.

Comparison of some early event-based optical flow methods [29], [30], [88] can be found in [31]. Yet, comparison was only carried out on flow fields generated by a rotating camera and so lacking motion parallax and occlusion (since an IMU was used to provide ground truth flow).

Opportunities: Comprehensive datasets with accurate ground truth optical flow in multiple scenarios (varying texture, speed, parallax, occlusions, illumination, etc.) and a common evaluation methodology would be essential to assess progress and reproducibility in this paramount low-level vision task. Section 6.2 outlines efforts in this direction, however, providing ground truth *event-based* optical flow in real scenes is challenging, especially for moving objects not conforming to the motion field induced by the camera's ego-motion. A thorough quantitative comparison of the event-based optical flow methods in the literature would help identify key ideas to further improve the methods.

4.3 3D reconstruction. Monocular and Stereo

Depth estimation with event cameras is a broad field. It can be divided according to the considered scenario and camera setup or motion, which determine the problem assumptions.

Instantaneous Stereo: Most works on depth estimation with event cameras target the problem of "instantaneous" stereo, i.e., 3D reconstruction using events on a very short time (ideally on a per-event basis) from two or more synchronized cameras that are rigidly attached. Being synchronized, the events from different image planes share a common clock. These works follow the classical two-step stereo solution: first solve the event correspondence problem across image planes (i.e., epipolar matching) and then triangulate the location of the 3D point [184]. Events are matched in two ways: (*i*) either using traditional stereo methods on artificial frames generated by accumulating events over time [114], [185] or generated using event timestamps [17] ("time surfaces"), or (*ii*) exploiting simultaneity and temporal correlations of the events across sensors [186], [187]. These approaches are *local*, matching events by comparing their neighborhoods since events cannot be matched based on individual timestamps. Additional constraints, such as the epipolar constraint [188], ordering, uniqueness, edge orientation and polarity may be used to reduce matching ambiguities, thus improving depth estimation [21], [151], [189]. Event matching can also be done by comparing local context descriptors [190], [191] of the spatial distribution of events on both stereo image planes.

Global approaches aim at getting smoother depth maps by considering smoothness constraints between neighboring points. In this category we find works, such as [22], [61], [145], [146], [192], that extend Marr and Poggio's cooperative stereo algorithm [143] to the case of event cameras. They use not only the temporal similarity to match events but also their spatio-temporal neighborhoods, with iterative nonlinear operations that result in an overall globally-optimal solution. Also in this class are [24], [193], which use belief propagation on a Markov Random Field or semiglobal matching [194] to improve stereo matching. A table comparing different stereo methods is provided in [25]; however, it should be interpreted with caution since the methods were not benchmarked on the same dataset.

Recently, brute-force space-sweeping using dedicated hardware (a GPU) has been proposed [195]. The method is based on ideas similar to [26], [196]: the correct depth manifests as "in focus" voxels of displaced events in the Disparity Space Image (DSI) [196], [197]. In contrast, other approaches pair event cameras with neuromorphic processors (Section 5.1) to produce low-power (100 mW), high-speed stereo systems [25]. There is an efficiency vs. accuracy trade-off that has not been quantified yet.

Most of the methods above are demonstrated in scenes with static cameras and few moving objects, so that event matches are easy to find due to uncluttered event data.

(a) (b)

Figure 6. Example of monocular depth estimation with a hand-held event camera. (a) Scene, (b) semi-dense depth map, pseudo-colored from red (close) to blue (far). Image courtesy of [23].

Event matching happens with low latency, at high rate (∼1 kHz) and consuming little power, which shows that event cameras are promising for high-speed 3D reconstructions of moving objects or in uncluttered scenes. Some companies, such as Prophesee, commercialize development kits that include some of the above-mentioned algorithms.

Multi-Perspective Panoramas: Some works [27], [198] also target the problem of instantaneous stereo (depth maps produced using events over very short time intervals), but using two non-simultaneous event cameras. These methods exploit a constrained hardware setup (two rotating event cameras with known motion) to either (*i*) recover intensity images on which conventional stereo is applied [198] or (*ii*) match events using temporal metrics [27].

Monocular Depth Estimation: Depth estimation with a single event camera has been shown in [23], [26], [47], [48], [196]. It is a significantly different problem from the above-mentioned ones because temporal correlation between events across multiple image planes cannot be exploited. These methods recover a semi-dense 3D reconstruction of the scene (i.e., a 3D edge map) by integrating information from the events of a moving camera over some time interval, and therefore, require camera motion information. Hence, these methods do not target the problem of instantaneous depth estimation, but rather the problem of depth estimation for visual odometry (VO) and SLAM [199].

The method in [47] is part of a pipeline that uses three filters operating in parallel to jointly estimate the motion of the event camera, a 3D map of the scene, and the intensity image. Their depth estimation approach requires using an additional quantity—the intensity image—to solve for data association. In contrast, [23], [196] proposes a space-sweep method that leverages directly the sparsity of the event stream to perform 3D reconstruction without having to establish event matches or recover the intensity images. It is computationally efficient and used for VO in [48].

Stereo Depth for SLAM: Recently, inspired by work in small-baseline multi-view stereo [200], a stereo depth estimation method for SLAM has been proposed [201]. It seeks to maximize the local spatio-temporal consistency of events across image planes using time surfaces [17].

Depth Estimation using Structured Light: All the above 3D reconstruction methods are passive, i.e., do not interfere with the scene. In contrast, there are some works on event-based active 3D reconstruction, based on emitting light onto the scene and measuring reflection with event cameras [28], [202], [203]. For example, [202] combines a DVS with a pulsed line laser to allow fast terrain reconstruction. Motion Contrast 3D scanning [28] is a structured light technique that simultaneously achieves high resolution, high speed and robust performance in challenging 3D scanning environments (e.g., strong illumination, or highly reflective and moving surfaces).

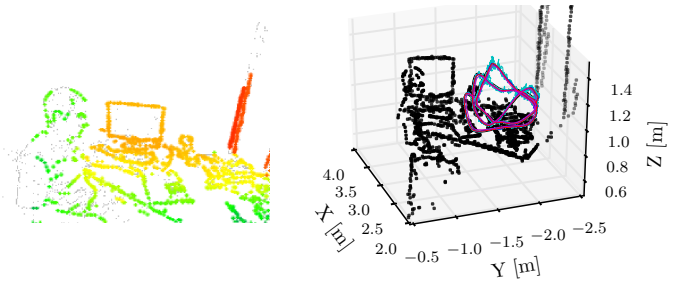
Table 3
Event-based methods for pose tracking and/or mapping with an event camera. The type of motion is noted with labels "2D" (3-DOF motions, e.g., planar or rotational) and "3D" (free 6-DOF motion in 3D space). Columns indicate whether the method performs tracking ("Track") and depth estimation ("Depth") using only events ("Event"), the type of scene considered ("Scene"), and any additional requirements. Only [47], [48] address the most general scenario using only events.

Reference	Dim	Track	Depth	Scene	Event	Additional requirements
Cook [35]	2D	✓	✗	natural	✓	rotational motion only
Weikersdorfer [44]	2D	✓	✗	B&W	✓	scene parallel to planar motion
Kim [39]	2D	✓	✗	natural	✓	rotational motion only
Gallego [116]	2D	✓	✗	natural	✓	rotational motion only
Reinbacher [204]	2D	✓	✗	natural	✓	rotational motion only
Censi [45]	3D	✓	✗	B&W	✗	attached depth sensor
Weikersdorfer [205]	3D	✓	✓	natural	✗	attached RGB-D sensor
Mueggler [42]	3D	✓	✗	B&W	✓	3D map of lines
Gallego [43]	3D	✓	✗	natural	✗	3D map of the scene
Rebecq [23], [196]	3D	✗	✓	natural	✓	pose information
Kueng [46]	3D	✓	✓	natural	✗	intensity images
Kim [47]	3D	✓	✓	natural	✓	image reconstruction
Rebecq [48]	3D	✓	✓	natural	✓	–

Opportunities: Although there are many methods for event-based depth estimation, it is difficult to compare their performance since they are not evaluated on the same dataset. In this sense, it would be desirable to (*i*) provide a comprehensive dataset for event-based depth evaluation (like effort [183]) and (*ii*) benchmark many existing methods on the dataset, to be able to compare their performance.

4.4 Pose Estimation and SLAM

Overview: Remarkable advances in event-based processing have been developed while addressing the problem of Simultaneous Localization and Mapping (SLAM). Since the event-based SLAM problem in its most general setting (6-DOF motion and natural 3D scenes) is a challenging problem, historically, it has been addressed step-by-step in scenarios with increasing complexity. Three complexity axes can be identified: dimensionality of the problem, type of motion and type of scene. The literature is dominated by methods that address the localization subproblem first (i.e., motion estimation) since it has fewer degrees of freedom to estimate and data association is easier than in the mapping subproblem. Regarding the type of motion, solutions for constrained motions, such as rotational or planar (both being 3-DOF), have been investigated before addressing the most complex case of a freely moving camera (6-DOF). Solutions for artificial scenes in terms of photometry (high contrast) and/or structure (line-based or 2D maps) have been proposed before focusing on the most difficult case: natural scenes (3D and with arbitrary photometric variations). Some proposed solutions require additional sensing (e.g., RGB-D) to reduce the complexity of the problem. This, however, introduces some of the bottlenecks present in frame-based systems (e.g., latency and motion blur). Table 3 classifies the related work using these complexity axes.



(a) Events & projected map. (b) Camera trajectory & 3D map.

Figure 7. Event-based SLAM. (a) Reconstructed scene from [206], with the reprojected semi-dense map colored according to depth and overlaid on the events (in gray), showing the good alignment between the map and the events. (b) Estimated camera trajectory (several methods) and semi-dense 3D map (i.e., point cloud). Image courtesy of [119].

Camera Tracking Methods: The first work on camera tracking with an event camera was presented in [41], and used a particle filter. The system was limited to slow planar motions and planar scenes parallel to the plane of motion consisting of artificial B&W line patterns. In [45], a standard grayscale camera was attached to a DVS to estimate, using a Bayesian filter, the small displacement between the current event and the previous frame of the standard camera. The system was developed for planar motion and B&W scenes. In [207], pose tracking under a non-holonomic and planar motion was proposed, supporting loop closure and topologically-correct trajectories.

Estimation of the 3D orientation of an event camera has been addressed in [35], [39], [116], [204]. Such systems are restricted to rotational motions, and, thus, do not account for translation or depth. Nevertheless they inspire ideas to solve more complex problems, such as [26], [47], [49], [50].

Regarding 6-DOF motion estimation, an event-based algorithm to track the pose of a DVS alone and during very high-speed motion was presented in [42]. The method was developed for artificial, B&W line-based maps. A continuous-time formulation of such method was given in [118] to estimate camera trajectory segments. In contrast, the event-based probabilistic filter in [43] showed 6-DOF high-speed tracking capabilities in natural scenes. Other pose tracking approaches have been developed as part of event-based SLAM systems, and are discussed next.

Tracking and Mapping: Cook et al. [35] proposed a generic message-passing algorithm within an interacting network to jointly estimate ego-motion, image intensity and optical flow from events. The idea of jointly estimating all these relevant quantities is appealing. However, the system was restricted to rotational motion.

An event-based 2D SLAM system was presented in [44] by extension of [41], and thus it was restricted to planar motion and high-contrast scenes. The method was extended to 3D in [205], but it relied on an external RGB-D sensor attached to the event camera for depth estimation.

The filter-based approach in [39] showed how to simultaneously track the 3D orientation of an event camera and create high-resolution panoramas of natural scenes. The system was limited to rotational motion, but popularized the idea that HDR intensity images can be recovered from the events. SLAM during rotational motion was also presented in [204], where camera tracking was done using direct registration methods on probabilistic edge maps [44].

Recently, solutions to the full problem of event-based 3D SLAM for 6-DOF motions and natural scenes, not relying on additional sensing, have been proposed [47], [48] (Table 3). The approach in [47] extends [39] and consists of three interleaved probabilistic filters to perform pose tracking as well as depth and intensity estimation. It is computationally intensive, requiring a GPU for real-time operation. In contrast, the semi-dense approach in [48] shows that intensity reconstruction is not needed for depth estimation (as proven in [23]) or pose tracking (edge-map alignment suffices). The resulting SLAM system runs in real-time on a CPU.

Taxonomy of Methods: From a methodology point of view, probabilistic filters have been the dominant paradigm to describe both tracking and mapping problems, e.g., [39], [41], [43], [44], [45], [47], [205]. They operate on an event-by-event basis, therefore asynchronously and with very low latency ($\sim$ µs), thus matching the characteristics of event cameras. However, updating the state of the system on a per-event basis can be computationally demanding, and so these methods typically require dedicated hardware (a GPU) to operate in real time. Trading off latency for efficiency, probabilistic filters can operate on small groups of events, simultaneously. Other approaches are natively designed in this way, based, for example, on non-linear optimization [26], [48], [116], [204], and run in real time on the CPU. Processing multiple events simultaneously is beneficial to reduce estimation noise.

Since events are caused by the apparent motion of intensity edges, it is not surprising to see that the majority of maps emerging from SLAM systems are maps of scene edges: the magnitude of gradient of the scene intensity is correlated with the spatial event firing rate of a map point, and it is enough to track reliably [23], [41], [48], [204], [205]. Intensity maps can also be used [39], [43], [47], from which spatial differences yield information of scene edges.

Opportunities: The above-mentioned SLAM methods lack loop-closure capabilities to reduce drift. Currently, the scales of the scenes on which event-based SLAM has been demonstrated are considerably smaller than those of frame-based SLAM. However, trying to match both scales may not be a sensible goal since event cameras may not be used to tackle the same problems as standard cameras; both sensors are complementary, as argued in [43], [45], [50], [160]. Stereo event-based SLAM is another unexplored topic, as well as designing more accurate, efficient and robust methods than the existing monocular ones.

4.5 Visual-Inertial Odometry (VIO)

The robustness of event-based visual odometry and SLAM systems can be improved by sensor fusion, e.g., by combining an event camera with an inertial measurement unit (IMU) rigidly attached. For this and other reasons, some event cameras have an integrated IMU (see Table 1).

Feature-based VIO: The majority of existing event-based VIO systems are "feature-based", consisting of two stages: first, features tracks are extracted from the events, and then these point trajectories on the image plane are fused with IMU measurements using state-of-the-art VIO

algorithms, such as [208], [209], [210]. For example, [120] tracked features using [117], and combined them with IMU data by means of the Kalman filter in [208]. Recently, [49] proposed to synthesize motion-compensated event images [116] and then detect-and-track features using classical methods [169], [172]. Feature tracks were fused with inertial data using keyframe-based nonlinear optimization [209] to recover the camera trajectory and a sparse map of 3D landmarks. The work in [49] was extended to fuse events, IMU data and standard intensity frames in [50], and was demonstrated on a computationally limited platform, such as a quadrotor, enabling it to fly in low light and HDR scenarios by exploiting the advantages of event cameras. The above methods are benchmarked on the 6-DOF motion dataset [206], and each method outperforms its predecessor.

Reprojection-error–based VIO: The work in [119] presents a different approach, fusing events and inertial data using a continuous-time framework [211]. As opposed to the above-mentioned feature-based methods, it optimizes a combined objective functional with inertial- and event-reprojection error terms over a segment of the camera trajectory, in the style of visual-inertial bundle adjustment.

Opportunities: The above works show that, although event cameras work very differently from standard cameras, it is possible to adapt state-of-the-art methods from multi-view computer vision [184] once events have been "converted" from photometric to geometric information (e.g., from events to feature tracks). However, it should be possible to avoid this conversion step and directly recover the camera motion and scene structure from the events, as suggested by [26]; for example, by optimizing a function with photometric (i.e., event firing rate [48]) and inertial error terms, akin to VI-DSO [212] for standard cameras.

Stereo event-based VIO is an unexplored topic, and it would be interesting to see how ideas from event-based depth estimation can be combined with SLAM and VIO.

Also to be explored are learning-based approaches to tackle all of the above problems. Currently, literature is dominated by model-based methods, but, as it happened in frame-based vision, we anticipate that learning-based methods will also play a major role in event-based processing. Some works in this direction are [17], [18], [33], [128], [130].

4.6 Image Reconstruction (IR)

Events represent brightness changes, and so, in ideal conditions (noise-free scenario, perfect sensor response, etc.) integration of the events yields "absolute" brightness. This is intuitive, since events are just a non-redundant per-pixel way of encoding the visual content in the scene. Moreover, due to the very high temporal resolution of the events, brightness images can be reconstructed at very high frame rate (e.g., 2 kHz [213]), or even continuously in time [37].

Literature Review: Image reconstruction (IR) from events was first established in [35], under rotational camera motions. A message-passing algorithm between pixels in a network of visual maps was used to jointly estimate several quantities, such as scene brightness. Also under rotational motion, [39] showed how to reconstruct high-resolution panoramas from the events, and they popularized the idea of even-based HDR image reconstruction. Each pixel of the panoramic image used a Kalman filter to estimate the brightness gradient, which was then integrated using Poisson reconstruction to yield absolute brightness. The method in [214] exploited the constrained motion of a platform rotating around a single axis to perform IR; the reconstructed images from were used for stereo depth estimation. In [32], IR was used as a means to aid the estimation of optical flow. Both image brightness and optical flow were jointly estimated using a variational framework to explain a space-time volume of events. Later, [36], [215] showed, using a variational image denoising approach, that IR was possible even without having to estimate the apparent motion (camera motion [35], [39] or optical flow [32]). Also without knowledge of the apparent motion, [213] performed IR; they used sparse signal processing with a patch-based learned dictionary that mapped events to image gradients, which were then Poisson-integrated. Recently, the VO methods in [47], [48] extended the IR technique in [39] to 6-DOF camera motions by using the computed scene depth and poses: [47] used a robust variational regularizer to reduce noise and improve contrast of the reconstructed image, whereas [48] showed IR as an ancillary result, since it was not needed to achieve VO. More recently, [37] proposed a per-pixel temporal smoothing filter for IR as well as to continuously fuse events and frames, and [38] presented a deep learning IR approach that achieved considerable gains over previous methods. Note that IR methods used in VO or SLAM [35], [39], [47] assume static scenes, whereas methods based on optical flow [32] or lack of motion information [36], [37], [38], [213], [215] work on dynamic scenes.

Figure 8. Image reconstruction example. Camera pointing at the Sun, in front of a traffic sign. Left: view from a standard camera, showing severe under-exposure on the foreground. Middle: frame from the DAVIS [4], showing severe under- and over-exposed areas. Right: HDR image reconstructed from the events. Image courtesy of [48].

Besides IR from events, another category of methods tackles the problem of fusing events and frames (e.g., from the DAVIS [4]), thus augmenting the brightness information from the frames with high temporal resolution and HDR properties of events. This is shown in [37], [40].

What Enables Image Reconstruction?: An interesting aspect of IR from events is that it requires some form of regularization. Event cameras have independent pixels that report brightness changes, and, consequently, per-pixel integration of such changes during a time interval only produces brightness increment images. To recover the absolute brightness at the end of the interval, an offset image would need to be added to the increment [40], [206]: the brightness image at the start of the interval. Surprisingly, [36], [213], [215] used spatial smoothing to perform IR starting from a

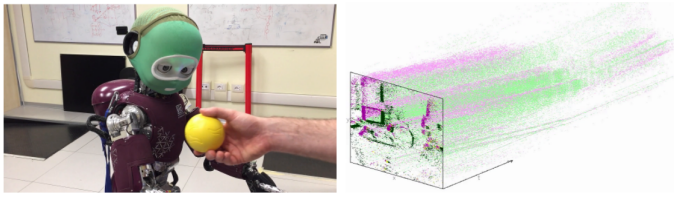


Figure 9. The iCub humanoid robot from IIT has two event cameras in the eyes. Here, it tracks a ball under event clutter produced by the motion of the head, and therefore, the event camera. Right: space-time visualization of the events on the image frame, colored according to polarity (positive in green, negative in red). Image courtesy of [12].

zero initial condition, i.e., without knowledge of the offset image. The temporal smoothing method in [37] uses an exponential kernel to wash out the effect ot the missing offset image. Other forms of regularization, using learned features from natural scenes [38], [213], [216] are also effective.

Image quality: The quality of the reconstructed image is directly affected by noise in the contrast threshold (Section 2.4), which changes per pixel (due to manufacturing mismatch) and also due to dynamical effects (incident light, time, etc.) [40]. Image quality has been quantified in several works [37], [38], [215], [216]. Image quality is also affected by the spatial resolution of the sensor.

Applications of Image Reconstruction: IR implies that, in principle, it is possible to convert the events into brightness images and then apply mature computer vision algorithms [38]. This can have a high impact on both, event- and frame-based communities. The resulting images capture high-speed motions and HDR scenes, which may be beneficial in some applications, but it comes at the expense of computational cost, latency and power consumption.

Despite IR having been useful to support tasks such as recognition [213], SLAM [47] or optical flow estimation [32], there are also works in the literature, such as [18], [26], [33], [48], [129], [130], showing that IR is not needed to fulfill such tasks. One of the most valuable aspect of IR is that it provides scene representations (e.g., appearance maps [39], [43]) that are more *invariant* to motion than events and also facilitate establishing event correspondences, which is one of the biggest challenges of event data processing [160].

4.7 Motion Segmentation

In Section 4.1 we distinguished between two scenarios: static or moving camera. In case of a static camera[10], events are caused by moving objects, hence, segmentation of such objects is trivial: it reduces to object detection by event activity. In this section, we review object segmentation in its non-trivial form: in the presence of event clutter, which is typically imputable to the apparent motion of the background due to the moving camera. Thus, events are caused by both, objects of interest as well as clutter, and the goal is to infer this classification for each event.

The work in [11] presents a method to detect and track a circle in the presence of event clutter caused by the moving camera. It is based on the Hough transform using optical flow information extracted from temporal windows of events. The method was extended in [12] using a particle filter to improve tracking robustness: the duration of the observation window was dynamically selected to accommodate for sudden motion changes due to accelerations of the object. Segmentation of an independently moving object with respect to event clutter was also addressed in [175]. It considered more generic object types by detecting and tracking event corners as primitives; and it used a learning technique to separate events caused by camera motion from those due to the object, based on additional knowledge of the robot joints controlling the camera.

Segmentation has also been addressed by exploiting the idea of motion-compensated (i.e., sharp) event images [116]. The method in [181] simultaneously estimates optical flow and segments the scene into objects that travel with distinct velocities. Thus, it clusters events according to optical flow, yielding motion-corrected images with sharp object contours. Similarly, [217] detects moving objects in clutter by fitting a motion-correction model to the dominant events (i.e., the background) and detecting inconsistencies with respect to that motion (i.e., the objects). They test their method in challenging scenarios inaccessible to standard cameras (HDR, high-speed) and release their dataset. More advanced works are [132], [218].

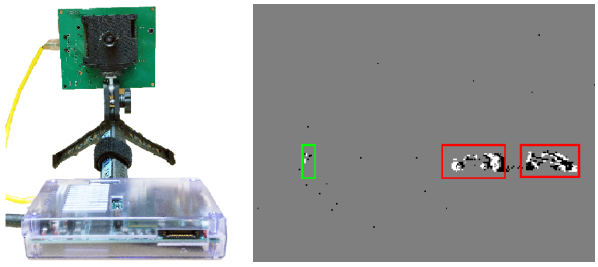
4.8 Recognition

Algorithms: Recognition algorithms for event cameras have grown in complexity, from template matching of simple shapes to classifying arbitrary edge patterns using either traditional machine learning on hand-crafted features or modern deep learning methods. This evolution aims at endowing recognition systems with more expressibility (i.e., approximation capacity) and robustness to data distortions.

Early research with event-based sensors began with tracking a moving object using a static sensor. An event-driven update of the position of a model of the object shape was used to detect and track objects with a known simple shape, such as a blob [8], circle [15], [71] or line [72]. Simple shapes can also be detected by matching against a predefined template, which removes the need to describe the geometry of the object. This *template matching* approach was implemented using convolutions in early hardware [71].

For more complex objects, templates can be used to match low level features instead of the entire object, after which a *classifier* can be used to make a decision based on the distribution of features observed [17]. Nearest Neighbor classifiers are typically used, with distances calculated in feature space. Accuracy can be improved by increasing feature invariance, which can be achieved using a hierarchical model where feature complexity increases in each layer. With a good choice of features, only the final classifier needs to be retrained when switching tasks. This leads to the problem of selecting which features to use. Hand-crafted orientation features were used in early works, but far better results are obtained by learning the features from the data itself. In the simplest case, each template can be obtained from an individual sample, but such templates are sensitive to noise in the sample data [16]. One may follow a generative approach, learning features that enable to accurately reconstruct the input, as was done in [108] with

10. We also assume constant illumination.



(a) (b)

Figure 10. IBM's TrueNorth neurosynaptic system performing recognition of moving objects. (a) A DAVIS240C sensor with FPGA attached performing tracking and sending tracked regions to IBM's TrueNorth NS1e evaluation platform for classification. (b) Tracking and classification results on a street scene. Red boxes indicate cars and the green box indicates a pedestrian.

a Deep Belief Network (DBN). More recent work obtains features by unsupervised learning, clustering the event data and using the center of each cluster as a feature [17]. During inference, each event is associated to its closest feature, and a classifier operates on the distributions of features observed. With the rise of *deep learning* in frame-based computer vision, many have sought to leverage deep learning tools for event-based recognition, using back-propagation to learn features. This approach has the advantage of not requiring a separate classifier at the output, but the disadvantage of requiring far more labeled data for training.

Most learning-based approaches convert events/spikes into (dense) tensors, a convenient representation for image-based hierarchical models, e.g., neural networks (Fig 10). There are different ways the value of each tensor element can be computed. Simple methods use the time since the last event at the corresponding pixel or neuron, or count the number of events received within a certain time period. A more robust method sets the tensor element to 1 whenever an event is received, and allows it to decay exponentially down to 0 over time [17], [18]. Image reconstruction methods (Section 4.6) may also be used. Some recognition approaches rely on converting spikes to frames during inference [125], [213], while others convert the trained artificial neural network to a spiking neural network (SNN) which can operate directly on the event data [107]. Similar ideas can be applied for tasks other than recognition [33], [128]. As neuromorphic hardware advances (see Section 5.1), there is increasing interest in learning directly in SNNs [153] or even directly in the neuromorphic hardware itself [219].

Tasks: Early tasks focused on detecting the presence of a simple shape (such as a circle) from a static sensor [8], [15], [71], but soon progressed to the classification of more complex shapes, such as card pips [107], block letters [16] and faces [17], [213]. A popular task throughout has been the classification of hand-written digits. Inspired by the role it has played in conventional frame-based computer vision, a few event-based MNIST datasets have been generated from the original MNIST dataset [77], [220]. These datasets remain a good test for algorithm development, with many algorithms now achieving over $98\,\%$ accuracy on the task [18], [112], [153], [221], [222], [223], but few would propose digit recognition as a strength of event-based vision. More difficult tasks involve either more difficult objects, such as the Caltech-101 and Caltech-256 datasets (both of which are still considered easy by computer vision) or more difficult scenarios, such as recognition from on-board a moving vehicle [18]. Very few works tackle these tasks so far, and those that do typically fall back on generating frames from events and processing them using a traditional deep learning framework.

A key challenge for recognition is that event cameras respond to relative motion in the scene (Section 2.3), and thus require either the object or the camera to be moving. It is therefore unlikely that event cameras will be a strong choice for recognizing static or slow moving objects, although little has been done to combine the advantages of frame- and event-based cameras for these applications. The event-based appearance of an object is highly dependent on the above-mentioned relative motion (Fig. 4), thus tight control of the camera motion could be used to aid recognition [220].

Since the camera responds to dynamic signals, obvious applications would include recognizing objects by the way they move [224], or recognizing dynamic movements such as gestures or actions [19], [20]. These tasks are typically more challenging than static object recognition because they include a time dimension, but this is exactly where event cameras excels.

Opportunities: Event cameras exhibit many alluring properties, but event-based recognition has a long way to go if it is to compete with modern frame-based approaches. While it is important to compare event- and frame-based methods, one must remember that each sensor has its own strengths. The ideal acquisition scenario for a frame-based sensor consists of both the sensor and object being static, which is the worst possible scenario for the event-based sensor. For event-based recognition to find widespread adoption, it will need to find applications which play to its strengths. Such applications are unlikely to be similar to well established computer vision recognition tasks which play to the frame-based sensor's strengths. Instead, such applications are likely to involve resource constrained recognition of dynamic sequences, or recognition from on-board a moving platform. Finding and demonstrating the use of event-based sensors in such applications remains an open challenge for the community.

Although event-based datasets have improved in quality in recent years, there is still room for improvement. Much more data is being collected, but annotation remains challenging. There is not yet an agreed upon or standard tool or format for annotations. Many event-based datasets are derived from frame-based vision. While these datasets have played an important role in the field, they inherently play to the strengths of frame-based vision and are thus unlikely to give rise to new event-based sensor applications. Data collection and annotation is a tiresome and thankless task, but developing an easy to use pipeline for collecting and annotating event-based data would be a significant contribution to the field, especially if the tools can mature to the stage where the task can be outsourced to laymen.

4.9 Neuromorphic Control

In living creatures, most information processing happens through spike-based representation: spikes encode the sen-

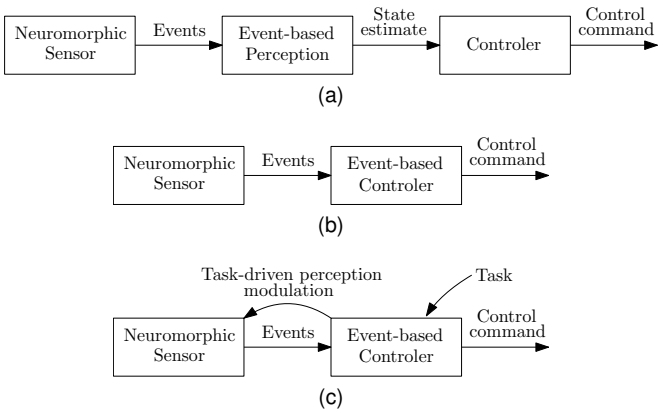


Figure 11. Control architectures based on neuromorphic events. In a neuromorphic-vision-driven control architecture (a), a neuromorphic sensor produces events, an event-based perception system produces state estimates, and a traditional controller is called asynchronously to compute the control signal. In a native neuromorphic-based architecture (b), the events generate directly changes in control. Finally, (c) shows an architecture in which the task informs the events that are generated.

sory data; spikes perform the computation; and spikes transmit actuator "commands". Therefore, biology shows that the event-based paradigm is, in principle, applicable not just to perception and inference, but also to control.

Neuromorphic-vision-driven Control Architecture: In this type of architecture (Fig. 11), there is a neuromorphic sensor, an event-based estimator, and a traditional controller. The estimator computes a state, and the controller computes the control based on the provided state. The controller is not aware of the asynchronicity of the architecture.

Neuromorphic-vision-driven control architectures have been demonstrated since the early days of neuromorphic cameras, and they have proved the two advantages of low latency and computational efficiency. An early example was the pencil-balancing robot [72]. In that demonstrator two DVS's observed a pencil as inverted pendulum placed on a small movable cart. The pencil's state in 3D was estimated in below $1\,\mathrm{ms}$ latency. A simple hand tuned PID controller kept the pencil balanced upright. It was also demonstrated on an embedded system, thereby establishing the ability to run on severely constrained computing resources. Another similar demonstrator was the "robot goalie" described in [8], [10], which showed similar principles.

Event-based Control Theory: Event-based techniques can be motivated from the perspective of control and decision theory. Using a biological metaphor, event-based control can be understood as a form of what economics calls *rational inattention* [225]: more information allows for better decisions, but if there are costs associated to obtaining or processing the information, it is rational to take decisions with only partial information available.

In event-based control, the control signal is changed asynchronously, that is, not synchronously with a reference clock [226]. There are several variations of the concept depending on how the "control events" are generated. One important distinction is between *event-triggered control* and *self-triggered control* [227]. In *event-based control* the events are generated "exogenously" based on certain condition; for example, a "recompute control" request might be triggered when the trajectory's tracking error exceeds a threshold. In *self-triggered control*, the controller decides by itself when is the next time it should be called based on the situation. For example, a controller might decide to "sleep" for longer if the state is near the target, or to recompute the control signal sooner if it is required.

The advantages of event-based control are usually justified considering a trade-off between computation / communication cost and control performance. The basic consideration is that, while the best control performance is obtained by recomputing the control infinitely often (for an infinite cost), there are strongly diminishing returns. A solid principle of control theory is that the control frequency depends on the time constant of the plant and the sensor: it does not make sense to change the control much quicker than the new incoming information or the speed of the actuators. This motivates choosing control frequencies that are comparable with the plant dynamics and adapt to the situation. For example, one can show that an event-triggered controller achieves the same performance with a fraction of the computation; or, conversely, a better performance with the same amount of computation. In some cases (scalar linear Gaussian) these trade-offs can be obtained in closed form [228], [229]. (Analogously, certain trade-offs can be obtained in closed form for perception [230].)

Unfortunately, the large literature in event-based control is of restricted utility for the embodied neuromorphic setting. Beyond the superficial similarity of dealing with "events" the settings are quite different. For example, in network-based control, one deals with typically low-dimensional states and occasional events—the focus is on making the most of each single event. By contrast, for an autonomous vehicle equipped with event cameras, the problem is typically how to find useful signals in potentially millions of events per second. Particularizing the event-based control theory to the neuromorphic case is a relatively young avenue of research [231], [232], [233], [234], [235]. The challenges lie in handling the non-linearities typical of the vision modality, which prevents clean closed-form results.

Open questions in Neuromorphic Control: Finally, we describe some of open problems in this topic.

Task-driven sensing: In animals, perception has value because it is followed by action, and the information collected is *actionable information* that helps with the task. A significant advance would be the ability for a controller to modulate the sensing process based on the task and the context. In current hardware there is limited software-modulated control for the sensing processing, though it is possible to modulate some of the hardware biases. Integration with region-of-interest mechanisms, heterogeneous camera bias settings, etc. would provide additional flexibility and more computationally efficient control.

Thinking fast and slow: Existing research has focused on obtaining low-latency control, but there has been little work on how to integrate this sensorimotor level into the rest of an agent's cognitive architecture. Using again a bio-inspired metaphor, and following Kahneman [236], the fast/instinctive/"emotional" system must be integrated with the slower/deliberative system.

Table 4
Comparison between selected neuromorphic processors, ordered by neuron model type.

	SpiNNaker [237]	TrueNorth [238]	Loihi [239]	DYNAP [240]	Braindrop [241]
Manufacturer	Univ. Manchester	IBM	Intel	aiCTX	Stanford Univ.
Neuron model	Software	Digital	Digital	Analog	Analog
On-chip learning	Yes	No	Yes	No	No
CMOS technology	130 nm	28 nm	14 nm	180 nm	28 nm
Year	2011	2014	2018	2017	2018
Neurons/chip	4 k*	1024 k	128 k	1 k	4 k
Neurons/core	255*	256	1024	256	4096
Cores/chip	16*	4096	128	4	1
Boards	4- or 48-chip	1- or 16-chip	4- or 8-chip,	1-chip	1-chip
Programming stack	sPyNNaker	CPE/Eedn	Nengo	cAER	Nengo
	PACMAN	NSCP	Nx SDK	libcAER	

5 Event-based Systems and Applications

5.1 Neuromorphic Computing

To build a vision system that is natively event-based from end to end, an event camera can be coupled with a neuromorphic processor. Neuromorphic engineering tries to capture some of the unparalleled computational power and efficiency of the brain by mimicking its structure and function. Typically this results in a massively parallel hardware accelerator for spiking neural networks, which is how we will define a neuromorphic processor. Since neuron spikes are inherently asynchronous events, a neuromorphic processor is the best possible computational partner for an event camera, and vice versa, because no overhead is required to convert frames into events or events into frames.

Neuromorphic processors may be categorized by their neuron model implementations (see Table 4), which are broadly divided between analog neurons like Neurogrid, BrainScaleS, ROLLS, and DYNAP-se; digital neurons like TrueNorth, Loihi, and ODIN; and software neurons like SpiNNaker. Some architectures also support on-chip learning (Loihi, ODIN, DYNAP-le).

When evaluating a neuromorphic processor for an event-based vision system, the following criteria should be considered in addition to the processor's functionality and performance:

- Any usable processor must be backed by a robust ecosystem of software development tools. A minimal toolchain includes an API to compose and train a network, a compiler to convert the network into a binary format that can be loaded into hardware, and a runtime library to deploy and operate the network in hardware.
- Event-based vision systems typically require that a processor be available as a standalone system suitable for mobile applications, and not just hosted in a remote server.
- Availability of the current generation of neuromorphic processors is entirely constrained by the manufacturer's limited capacity to supply and maintain them. There is almost always an early access program to provide hardware to selected research partners, and this may be the only way to get hardware, so price is typically not relevant.

Architectures

The following processors (Table 4) have the most mature developer workflows, combined with the widest availability of standalone systems.

SpiNNaker (Spiking Neural Network Architecture): Created at the University of Manchester, it uses general-purpose ARM cores to simulate biologically realistic models of the human brain. Unlike the other processors in Table 4, which all choose specific simplified neuron models to embed in custom transistor circuits, SpiNNaker implements neurons as software running on the ARM cores, sacrificing hardware acceleration to maximize model flexibility.

TrueNorth: The TrueNorth neurosynaptic processor from IBM uses digital neurons to perform real-time inference. Each chip simulates 1 million spiking neurons and 256 million synapses, distributed among 4096 neurosynaptic cores. There is no on-chip learning, so networks are trained offline using a GPU or other processor [242].

Examples of event-based vision systems that incorporate TrueNorth include a real-time gesture-recognition system that identifies ten different hand gestures from events acquired by a Samsung DVS-Gen2 camera [19], and a stereo vision application that reconstructs the distance to a moving object from the disparity between events from two iniVation DVS128 cameras [25].

Loihi: The Loihi spiking-neural-network chip from Intel uses digital neurons to perform real-time inference and online learning. Each chip simulates up to 128 thousand spiking neurons and up to 128 million synapses, distributed among 128 neuromorphic cores. A learning engine in each neuromorphic core uses filtered spike traces to update each synapse using a programmable selection from a set of rules that includes spike-timing-dependent plasticity (STDP) and reinforcement learning [239]. Non-spiking networks can be trained in TensorFlow and converted into approximately equivalent spiking networks for Loihi using the Nengo Deep Learning toolkit from Applied Brain Research [243].

DYNAP: The Dynamic Neuromorphic Asynchronous Processor (DYNAP) from aiCTX comes in two variants, one optimized for scalable inference (Dynap-se), and the other for online learning (Dynap-le).

Braindrop: Braindrop is Stanford University's follow-on to the Neurogrid processor. It is intended to prototype a single core of the planned 1-million-neuron Brainstorm system [241]. It is programmed using Nengo, and implements the Neural Engineering Framework (NEF).

(a) (b)

Figure 12. (a) Embedded DVS128 on Pushbot as standalone closed-loop perception-computation-action system, used in navigation and obstacle-avoidance tasks [244]. (b) Drone with downward-looking event camera, used for autonomous flight [50].

5.2 Applications in Real-Time On-Board Robotics

As event-based vision sensors often produce significantly less data per time interval compared to traditional cameras, multiple applications can be envisioned where extracting relevant vision information can happen in real-time within a simple computing system directly connected to the sensor, avoiding USB connection. Fig 12 shows an example of such, where a dual-core ARM micro controller running at 200 MHz with 136 kB on-board SRAM fetches and processes events in real-time. The combined embedded system of sensor and micro controller here operate a simple wheeled robot in tasks such as line following, active and passive object tracking, distance estimation, and up to simple environmental mapping [244].

A different example of near-sensor processing is the recently developed Speck SoC[11], which combines a DVS and the Dynapse Neuromorphic Convolutional Neuronal Network Processor. Application domains are low-power, continuous object detection, surveillance, and automotive systems.

Event cameras have also been used on-board quadrotors with limited computational resources, both for autonomous landing [245] or flight [50] (Fig. 12b), even in challenging conditions.

6 Resources

The List of Event-based Vision Resources [246] is a collaborative effort to collect relevant resources for event-based vision research. It includes not only links to information (papers, videos, organizations, companies and workshops) but also links to drivers, code, datasets, simulators and other essential tools in the field.

6.1 Software

To date, there is no open-source standard library integrated to OpenCV that provides algorithms for event-based vision. This would be a very desirable resource to accelerate the adoption of event cameras. There are, however, many quite highly developed open-source software resources:

- *jAER* [247][12] is a Java-based environment for event sensors and processing like noise reduction, feature extraction, optical flow, de-rotation using IMU, CNN and RNN inference, etc. Several non-mobile robots [8], [10], [72], [248] and even one mobile DVS [125] robot have been built in jAER, although Java is not ideal for mobile robots. It provides a desktop GUI based interface for easily recording and playing data that also exposes the complex internal configurations of these devices. It mainly supports the sensors developed at the Institute of Neuroinformatics (INI) of UZH-ETH Zurich that are distributed by iniVation.
- *libcaer*[13] is a minimal C library to access, configure and get data from iniVation and aiCTX neuromorphic sensors and processors. It supports the DVS and DAVIS cameras, and the Dynap-SE neuromorphic processor.
- *cAER*[14] is a very efficient event-based processing framework for neuromorphic devices, written in C/C++ for Linux, targeting embedded systems.
- The ROS DVS package[15] developed in the Robotics and Perception Group of UZH-ETH Zurich is based on libcaer. It provides C++ drivers for the DVS and DAVIS. It is popular in robotics since it integrates with the Robot Operating System (ROS) [249] and therefore provides high-level tools for easily recording and playing data, connecting to other sensors and actuators, etc. Popular datasets [183], [206] are provided in this format. The package also provides a calibration tool for both intrinsic and stereo calibration.
- The event-driven YARP Project[16] [250] comprises libraries to handle neuromorphic sensors, such as the DVS, installed on the iCub humanoid robot, along with algorithms to process event data. It is based on the Yet Another Robot Platform (YARP) middleware.
- pyAER[17] is a python wrapper around libcaer developed at the Dept. of Neuroinformatics (UZH-ETH) that will probably become popular for rapid experimentation.

Other open-source software utilities and processing algorithms (in Python, CUDA, Matlab, etc.) are spread throughout the web, on the pages of the research groups working on event-based vision [246]. Proprietary software includes the development kits (SDKs) developed by companies such as Prophesee, Samsung, Insightness or SLAMcore.

6.2 Datasets and Simulators

Datasets and simulators are fundamental tools to facilitate adoption of event-driven technology and advance its research. They allow to reduce costs (currently, event cameras are considerably more expensive than standard cameras) and to monitor progress with quantitative benchmarks (as in traditional computer vision: the case of datasets such as Middlebury, MPI Sintel, KITTI, EuRoC, etc.).

The number of event-based vision datasets and simulators is growing. Several of them are listed in [246], sorted by task. Broadly, they can be categorized as those that target motion estimation (regression) tasks and those that target recognition (classification) tasks. In the first group, there are datasets for optical flow, SLAM, object tracking, segmentation, etc. The second group comprises datasets for object and action recognition.

11. https://www.speck.ai/
12. https://jaerproject.org
13. https://github.com/inilabs/libcaer
14. https://github.com/inilabs/caer
15. https://github.com/uzh-rpg/rpg_dvs_ros
16. https://github.com/robotology/event-driven
17. https://github.com/duguyue100/pyaer

Datasets for optical flow include [31], [33], [251]. Since ground-truth optical flow is difficult to acquire, [31] considers only flow during purely rotational motion recorded with an IMU, and so, the dataset lacks flow due to translational (parallax) motion. The datasets in [33], [251] provide optical flow as the motion field induced on the image plane by the camera motion and the depth of the scene (measured with a range sensor, such as an RGB-D camera, a stereo pair or a LiDAR). Naturally, ground truth optical flow is subject to noise and inaccuracies in alignment and calibration of the different sensors involved.

Datasets for pose estimation and SLAM include [205][18], [183], [206], [251], [252]. The most popular one is described in [206], which has been used to benchmark visual odometry and visual-inertial odometry methods [26], [49], [50], [116], [119], [120], [204]. This dataset is also popular to evaluate corner detectors [123], [124] and feature trackers [46], [160].

Datasets for recognition are currently of limited size compared to traditional computer vision ones. They consist of cards of a deck (4 classes), faces (7 classes), handwritten digits (36 classes), gestures (rocks, papers, scissors) in dynamic scenes, cars, etc. Neuromorphic versions of popular traditional computer vision datasets, such as MNIST and Caltech101, have been obtained by using saccade-like motions [220], [253]. These datasets have been used in [16], [17], [18], [107], [125], [126], among others, to benchmark event-based recognition algorithms.

The DVS emulator in [83] and the simulator in [206] are based on the operation principle of an ideal DVS pixel (2). Given a virtual 3D scene and the trajectory of a moving DAVIS within it, the simulator generates the corresponding stream of events, intensity frames and depth maps. The simulator has been extended in [84], using an adaptive sampling-rendering scheme, being more photo-realistic, including an event noise model and also returning ground truth optical flow.

A comprehensive characterization of the noise and dynamic effects of existing event cameras has not been carried out yet, and so, the noise models used are, currently, only a coarse approximation. In the future, it would be desirable to develop more realistic sensor models so that prototyping on simulated data transferred more easily to real data.

6.3 Workshops

To date, there are two yearly Summer schools fostering research, among other topics, on event-based vision: the Telluride Neuromorphic Cognition Engineering Workshop (26th edition in 2019, in USA) and the Capo Caccia Cognitive Neuromorphic Engineering Workshop (11th edition in 2019, in Europe). Recently, workshops have been organized alongside major robotics conferences (IROS'15 Workshop on Innovative Sensing for Robotics[19], the ICRA'17 First International Workshop on Event-based Vision[20] or the IROS'18 Workshop on Unconventional Sensing and Processing for Robotic Visual Perception[21]). Live demos of event-based systems have been shown at top-tier conferences, such as ISSCC'06, NIPS'09, CVPR'18, ECCV'18, ICRA'17, IROS'18, multiple ISCAS, etc. As the event-based vision community grows, more workshops and live demonstrations are expected to happen also in traditional computer vision venues, such as CVPR'19[22].

7 Discussion

Event-based vision is a topic that spans many fields, such as computer vision, robotics and neuromorphic engineering. Each community focuses on exploiting different advantages of the event-based paradigm. Some focus on the low power consumption for "always on" or embedded applications on resource-constrained platforms; others favor low latency to enable highly reactive systems, and others prefer the availability of information to better perceive the environment (high temporal resolution and HDR), with fewer constraints on computational resources.

Event-based vision is an emerging technology in the era of mature frame-based camera hardware and software. Comparisons are, in some terms, unfair since they are not carried out under the same maturity level. Nevertheless event cameras show potential, able to overcome some of the limitations of frame-based cameras, reaching new scenarios previously inaccessible. There is considerable room for improvement (research and development), as pointed out in numerous opportunities throughout the paper.

There is no agreement on what is the best method to process events, notably because it depends on the application. There are different trade-offs involved, such as latency vs. power consumption and accuracy, or sensitivity vs. bandwidth and processing capacity. For example, reducing the contrast threshold and/or increasing the resolution produces more events, which will be processed by an algorithm and platform with finite capacity. A challenging research area is to quantify such trade-offs and to develop techniques to dynamically adjust the sensor and/or algorithm parameters for optimal performance.

Another big challenge is to develop bio-inspired systems that are natively event-based end-to-end (from perception to control and actuation) that are also more efficient and long-term solutions than synchronous, frame-based systems. Event cameras pose the challenge of rethinking perception, control and actuation, and, in particular, the current main stream of deep learning methods in computer vision: adapting them or transferring ideas to process events while being as top-performing. Active vision (pairing perception and control) is specially relevant on event cameras because the events distinctly depends on motion, which may be due to the actuation of a robot.

Event cameras can be seen as an entry point for more efficient, near-sensor processing, such that only high-level, non-redundant information is transmitted, thus reducing bandwidth, latency and power consumption. This could be done by pairing an event camera with hardware on the same sensor device (Speck in Section 5.2), or by alternative bio-inspired imaging sensors, such as cellular processor arrays [254] which every pixel has a processor that allows to perform several types of computations with the brightness of the pixel and its neighbors.

18. http://ebvds.neurocomputing.systems
19. http://innovative-sensing.mit.edu/
20. http://rpg.ifi.uzh.ch/ICRA17_event_vision_workshop.html
21. http://www.jmartel.net/irosws-home
22. http://rpg.ifi.uzh.ch/CVPR19_event_vision_workshop.html

8 Conclusion

Event cameras are revolutionary sensors that offer many advantages over traditional, frame-based cameras, such as low latency, low power, high speed and high dynamic range. Hence, they have a large potential for computer vision and robotic applications in challenging scenarios currently inaccessible to traditional cameras. We have provided an overview of the field of event-based vision, covering perception, computing and control, with a focus on the working principle of event cameras and the algorithms developed to unlock their outstanding properties in selected applications, from low-level vision to high-level vision. Neuromorphic perception and control are emerging topics; and so, there are plenty of opportunities, as we have pointed out throughout the text. Many challenges remain ahead, and we hope that this paper provides an introductory exposition of the topic, as a step in humanity's longstanding quest to build intelligent machines endowed with a more efficient, bio-inspired way of perceiving and interacting with the world.

References

[1] M. Mahowald and C. Mead, "The silicon retina," *Scientific American*, vol. 264, no. 5, pp. 76–83, May 1991.

[2] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[3] 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," *IEEE J. Solid-State Circuits*, vol. 46, no. 1, pp. 259–275, Jan. 2011.

[4] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, "A 240x180 130dB 3us latency global shutter spatiotemporal vision sensor," *IEEE J. Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, 2014.

[5] B. Son, Y. Suh, S. Kim, H. Jung, J.-S. Kim, C. Shin, K. Park, K. Lee, J. Park, J. Woo, Y. Roh, H. Lee, Y. Wang, I. Ovsiannikov, and H. Ryu, "A 640x480 dynamic vision sensor with a 9um pixel and 300Meps address-event representation," in *IEEE Intl. Solid-State Circuits Conf. (ISSCC)*, 2017.

[6] T. Delbruck, "Neuromorphic vision sensing and processing," in *Eur. Solid-State Device Research Conf. (ESSDERC)*, 2016, pp. 7–14.

[7] S.-C. Liu, B. Rueckauer, J. Anumula, A. Huber, D. Neil, and T. Delbruck, "Event-Driven Sensing for Efficient Perception," *IEEE Signal Process. Mag.*, 2019, (Under review).

[8] T. Delbruck and P. Lichtsteiner, "Fast sensory motor control based on event-based hybrid neuromorphic-procedural system," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2007, pp. 845–848.

[9] D. Drazen, P. Lichtsteiner, P. Häfliger, T. Delbrück, and A. Jensen, "Toward real-time particle tracking using an event-based dynamic vision sensor," *Experiments in Fluids*, vol. 51, no. 5, pp. 1465–1469, 2011.

[10] T. Delbruck and M. Lang, "Robotic goalie with 3ms reaction time at 4% CPU load using event-based dynamic vision sensor," *Front. Neurosci.*, vol. 7, p. 223, 2013.

[11] A. Glover and C. Bartolozzi, "Event-driven ball detection and gaze fixation in clutter," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2016, pp. 2203–2208.

[12] ——, "Robust visual tracking with a freely-moving event camera," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2017, pp. 3769–3776.

[13] M. Litzenberger, A. N. Belbachir, N. Donath, G. Gritsch, H. Garn, B. Kohn, C. Posch, and S. Schraml, "Estimation of vehicle speed based on asynchronous data from a silicon retina optical sensor," in *IEEE Intell. Transp. Sys. Conf.*, 2006, pp. 653–658.

[14] E. Piatkowska, A. N. Belbachir, S. Schraml, and M. Gelautz, "Spatiotemporal multiple persons tracking using dynamic vision sensor," in *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, 2012, pp. 35–40.

[15] G. Wiesmann, S. Schraml, M. Litzenberger, A. N. Belbachir, M. Hofstatter, and C. Bartolozzi, "Event-driven embodied system for feature extraction and object recognition in robotic applications," in *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, 2012, pp. 76–82.

[16] G. Orchard, C. Meyer, R. Etienne-Cummings, C. Posch, N. Thakor, and R. Benosman, "HFirst: A temporal approach to object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 2028–2040, 2015.

[17] X. Lagorce, G. Orchard, F. Gallupi, B. E. Shi, and R. Benosman, "HOTS: A hierarchy of event-based time-surfaces for pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1346–1359, Jul. 2017.

[18] A. Sironi, M. Brambilla, N. Bourdis, X. Lagorce, and R. Benosman, "HATS: Histograms of averaged time surfaces for robust event-based object classification," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 1731–1740.

[19] A. Amir, B. Taba, D. Berg, T. Melano, J. McKinstry, C. D. Nolfo, T. Nayak, A. Andreopoulos, G. Garreau, M. Mendoza, J. Kusnitz, M. Debole, S. Esser, T. Delbruck, M. Flickner, and D. Modha, "A low power, fully event-based gesture recognition system," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 7388–7397.

[20] J. H. Lee, T. Delbruck, M. Pfeiffer, P. K. Park, C.-W. Shin, H. Ryu, and B. C. Kang, "Real-time gesture interface based on event-driven processing from stereo silicon retinas," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2250–2263, 2014.

[21] P. Rogister, R. Benosman, S.-H. Ieng, P. Lichtsteiner, and T. Delbruck, "Asynchronous event-based binocular stereo matching," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 2, pp. 347–353, 2012.

[22] E. Piatkowska, A. N. Belbachir, and M. Gelautz, "Asynchronous stereo vision for event-driven dynamic stereo sensor using an adaptive cooperative approach," in *Int. Conf. Comput. Vis. Workshops (ICCVW)*, 2013, pp. 45–50.

[23] H. Rebecq, G. Gallego, E. Mueggler, and D. Scaramuzza, "EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time," *Int. J. Comput. Vis.*, vol. 126, no. 12, pp. 1394–1414, Dec. 2018.

[24] Z. Xie, S. Chen, and G. Orchard, "Event-based stereo depth estimation using belief propagation," *Front. Neurosci.*, vol. 11, Oct. 2017.

[25] A. Andreopoulos, H. J. Kashyap, T. K. Nayak, A. Amir, and M. D. Flickner, "A low power, high throughput, fully event-based stereo system," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 7532–7542.

[26] G. Gallego, H. Rebecq, and D. Scaramuzza, "A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 3867–3876.

[27] S. Schraml, A. N. Belbachir, and H. Bischof, "Event-driven stereo matching for real-time 3D panoramic vision," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2015, pp. 466–474.

[28] N. Matsuda, O. Cossairt, and M. Gupta, "MC3D: Motion contrast 3D scanning," in *IEEE Int. Conf. Comput. Photography (ICCP)*, 2015, pp. 1–10.

[29] R. Benosman, S.-H. Ieng, C. Clercq, C. Bartolozzi, and M. Srinivasan, "Asynchronous frameless event-based optical flow," *Neural Netw.*, vol. 27, pp. 32–37, 2012.

[30] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 407–417, 2014.

[31] B. Rueckauer and T. Delbruck, "Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor," *Front. Neurosci.*, vol. 10, no. 176, 2016.

[32] P. Bardow, A. J. Davison, and S. Leutenegger, "Simultaneous optical flow and intensity estimation from an event camera," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2016, pp. 884–892.

[33] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "EV-FlowNet: Self-supervised optical flow estimation for event-based cameras," in *Robotics: Science and Systems (RSS)*, 2018.

[34] F. Paredes-Valles, K. Y. W. Scheper, and G. C. H. E. de Croon, "Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception," *IEEE Trans. Pattern Anal. Mach. Intell.*, Mar. 2019.

[35] M. Cook, L. Gugelmann, F. Jug, C. Krautz, and A. Steger, "In-

teracting maps for fast visual interpretation," in *Int. Joint Conf. Neural Netw. (IJCNN)*, 2011, pp. 770–776.

[36] C. Reinbacher, G. Graber, and T. Pock, "Real-time intensity-image reconstruction for event cameras using manifold regularisation," in *British Mach. Vis. Conf. (BMVC)*, 2016.

[37] C. Scheerlinck, N. Barnes, and R. Mahony, "Continuous-time intensity estimation using event cameras," in *Asian Conf. Comput. Vis. (ACCV)*, 2018.

[38] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza, "Events-to-video: Bringing modern computer vision to event cameras," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.

[39] H. Kim, A. Handa, R. Benosman, S.-H. Ieng, and A. J. Davison, "Simultaneous mosaicing and tracking with an event camera," in *British Mach. Vis. Conf. (BMVC)*, 2014.

[40] C. Brandli, L. Muller, and T. Delbruck, "Real-time, high-speed video decompression using a frame- and event-based DAVIS sensor," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2014, pp. 686–689.

[41] D. Weikersdorfer and J. Conradt, "Event-based particle filtering for robot self-localization," in *IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, 2012, pp. 866–870.

[42] E. Mueggler, B. Huber, and D. Scaramuzza, "Event-based, 6-DOF pose tracking for high-speed maneuvers," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2014, pp. 2761–2768.

[43] G. Gallego, J. E. A. Lund, E. Mueggler, H. Rebecq, T. Delbruck, and D. Scaramuzza, "Event-based, 6-DOF camera tracking from photometric depth maps," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 10, pp. 2402–2412, Oct. 2018.

[44] D. Weikersdorfer, R. Hoffmann, and J. Conradt, "Simultaneous localization and mapping for event-based vision systems," in *Int. Conf. Comput. Vis. Syst. (ICVS)*, 2013, pp. 133–142.

[45] A. Censi and D. Scaramuzza, "Low-latency event-based visual odometry," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014, pp. 703–710.

[46] B. Kueng, E. Mueggler, G. Gallego, and D. Scaramuzza, "Low-latency visual odometry using event-based feature tracks," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2016, pp. 16–23.

[47] H. Kim, S. Leutenegger, and A. J. Davison, "Real-time 3D reconstruction and 6-DoF tracking with an event camera," in *Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 349–364.

[48] H. Rebecq, T. Horstschäfer, G. Gallego, and D. Scaramuzza, "EVO: A geometric approach to event-based 6-DOF parallel tracking and mapping in real-time," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 593–600, 2017.

[49] H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization," in *British Mach. Vis. Conf. (BMVC)*, 2017.

[50] A. Rosinol Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high speed scenarios," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 994–1001, Apr. 2018.

[51] L. Pan, C. Scheerlinck, X. Yu, R. Hartley, M. Liu, and Y. Dai, "Bringing a blurry frame alive at high frame-rate with an event camera," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.

[52] G. Cohen, S. Afshar, and A. van Schaik, "Approaches for astrometry using event-based sensors," in *Conf. Advanced Maui Optical and Space Surveillance Technologies*, 2017.

[53] P. Lichtsteiner and T. Delbruck, "64x64 event-driven logarithmic temporal derivative silicon retina," in *IEEE Workshop on Charge-Coupled Devices and Advanced Image Sensors*, 2005, pp. 157–160.

[54] ——, "A 64x64 AER logarithmic temporal derivative silicon retina," in *Research in Microelectronics and Electronics, PhD*, vol. 2, 2005, pp. 202–205.

[55] P. Lichtsteiner, "An AER temporal contrast vision sensor," Ph. D. Thesis, ETH Zurich, Dept. of Physics (D-PHYS), Zurich, Switzerland, 2006.

[56] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128x128 120dB 30mW asynchronous vision sensor that responds to relative intensity change," in *IEEE Intl. Solid-State Circuits Conf. (ISSCC)*, 2006, pp. 2060–2069.

[57] R. Berner, C. Brandli, M. Yang, S.-C. Liu, and T. Delbruck, "A 240x180 10mw 12us latency sparse-output vision sensor for mobile applications," in *Proc. Symp. VLSI*, 2013, pp. C186–C187.

[58] D. Neil, "Deep neural networks and hardware systems for event-driven data," Ph.D. dissertation, ETH-Zurich, Zurich, Switzerland, 2017.

[59] K. A. Boahen, "A burst-mode word-serial address-event link-I: Transmitter design," *IEEE Trans. Circuits Syst. I*, vol. 51, no. 7, pp. 1269–1280, Jul. 2004.

[60] S.-C. Liu, T. Delbruck, G. Indiveri, A. Whatley, and R. Douglas, *Event-Based Neuromorphic Systems*. John Wiley & Sons, 2015.

[61] M. Mahowald, "VLSI analogs of neuronal visual processing: A synthesis of form and function," Ph.D. dissertation, California Institute of Technology, Pasadena, California, May 1992.

[62] T. Delbruck and C. A. Mead, "Time-derivative adaptive silicon photoreceptor array," in *Proc. SPIE, Infrared sensors: Detectors, Electron., and Signal Process.*, vol. 1541, 1991, pp. 92–99.

[63] S.-C. Liu and T. Delbruck, "Neuromorphic sensory systems," *Current Opinion in Neurobiology*, vol. 20, no. 3, pp. 288–295, 2010.

[64] T. Delbruck, B. Linares-Barranco, E. Culurciello, and C. Posch, "Activity-driven, event-based vision sensors," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2010, pp. 2426–2429.

[65] T. Delbruck, "Fun with asynchronous vision sensors and processing," in *Eur. Conf. Comput. Vis. Workshops (ECCVW)*, 2012, pp. 506–515.

[66] C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, and T. Delbruck, "Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output," *Proc. IEEE*, vol. 102, no. 10, pp. 1470–1484, Oct. 2014.

[67] C. Posch, D. Matolin, and R. Wohlgenannt, "A QVGA 143dB dynamic range asynchronous address-event PWM dynamic image sensor with lossless pixel-level video compression," in *IEEE Intl. Solid-State Circuits Conf. (ISSCC)*, 2010, pp. 400–401.

[68] G. Orchard, D. Matolin, X. Lagorce, R. Benosman, and C. Posch, "Accelerated frame-free time-encoded multi-step imaging," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2014, pp. 2644–2647.

[69] E. R. Fossum, "CMOS image sensors: electronic camera-on-a-chip," *IEEE Trans. Electron Devices*, vol. 44, no. 10, pp. 1689–1698, Oct. 1997.

[70] M. Guo, J. Huang, and S. Chen, "Live demonstration: A 768 x 640 pixels 200meps dynamic vision sensor," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2017.

[71] 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, S.-C. 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-processing-learning-actuating system for high-speed visual object recognition and tracking," *IEEE Trans. Neural Netw.*, vol. 20, no. 9, pp. 1417–1438, 2009.

[72] J. Conradt, M. Cook, R. Berner, P. Lichtsteiner, R. J. Douglas, and T. Delbruck, "A pencil balancing robot using a pair of AER dynamic vision sensors," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2009, pp. 781–784.

[73] J. Conradt, "On-board real-time optic-flow for miniature event-based vision sensors," in *IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, 2015, pp. 1858–1863.

[74] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 3530–3538.

[75] Y. Nozaki and T. Delbruck, "Temperature and parasitic photocurrent effects in dynamic vision sensors," *IEEE Trans. Electron Devices*, vol. 64, no. 8, pp. 3239–3245, Aug. 2017.

[76] ——, "Authors Reply to Comment on Temperature and Parasitic Photocurrent Effects in Dynamic Vision Sensors," *IEEE Trans. Electron Devices*, vol. 65, no. 7, pp. 3083–3083, Jul. 2018.

[77] T. Serrano-Gotarredona and B. Linares-Barranco, "A 128 $\times$128 1.5contrast sensitivity 0.9dynamic vision sensor using transimpedance preamplifiers," *IEEE J. Solid-State Circuits*, vol. 48, no. 3, pp. 827–838, Mar. 2013.

[78] M. Yang, S.-C. Liu, and T. Delbruck, "A dynamic vision sensor with 1% temporal contrast sensitivity and in-pixel asynchronous delta modulator for event encoding," *IEEE J. Solid-State Circuits*, vol. 50, no. 9, pp. 2149–2160, 2015.

[79] D. P. Moeys, F. Corradi, C. Li, S. A. Bamford, L. Longinotti, F. F. Voigt, S. Berry, G. Taverni, F. Helmchen, and T. Delbruck, "A sensitive dynamic and active pixel vision sensor for color or neural imaging applications," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 123–136, Feb. 2018.

[80] A. Rose, *Vision: Human and Electronic*. Plenum Press, New York, 1973.

[81] C. Scheerlinck, N. Barnes, and R. Mahony, "Asynchronous spatial image convolutions for event cameras," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 816–822, Apr. 2019.

[82] G. Gallego, C. Forster, E. Mueggler, and D. Scaramuzza, "Event-based camera pose tracking using a generative event model," 2015, arXiv:1510.01972.

[83] M. L. Katz, K. Nikolic, and T. Delbruck, "Live demonstration: Behavioural emulation of event-based vision sensors," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2012, pp. 736–740.

[84] H. Rebecq, D. Gehrig, and D. Scaramuzza, "ESIM: an open event camera simulator," in *Conf. on Robotics Learning (CoRL)*, 2018.

[85] M. Yang, S.-C. Liu, and T. Delbruck, "Analysis of encoding degradation in spiking sensors due to spike delay variation," *IEEE Trans. Circuits Syst. I*, vol. 64, no. 1, pp. 145–155, Jan. 2017.

[86] C. Posch and D. Matolin, "Sensitivity and uniformity of a 0.18um CMOS temporal contrast pixel array," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2011, pp. 1572–1575.

[87] M. Yang, S.-C. Liu, and T. Delbruck, "Comparison of spike encoding schemes in asynchronous vision sensors: Modeling and design," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2014, pp. 2632–2635.

[88] T. Delbruck, "Frame-free dynamic digital vision," in *Proc. Int. Symp. Secure-Life Electron.*, 2008, pp. 21–26.

[89] A. Khodamoradi and R. Kastner, "O(N)-space spatiotemporal filter for reducing noise in neuromorphic vision sensors," *IEEE Trans. Emerg. Topics Comput.*, vol. PP, no. 99, pp. 1–1, 2018.

[90] D. Czech and G. Orchard, "Evaluating noise filtering for event-based asynchronous change detection image sensors," in *IEEE Int. Conf. Biomed. Robot. and Biomechatron. (BioRob)*, 2016, pp. 19–24.

[91] V. Padala, A. Basu, and G. Orchard, "A noise filtering algorithm for event-based asynchronous change detection image sensors on TrueNorth and its implementation on TrueNorth," *Front. Neurosci.*, vol. 12, 2018.

[92] https://www.prophesee.ai/event-based-evk/, 2019.

[93] T. Delbruck, V. Villanueva, and L. Longinotti, "Integration of dynamic vision sensor with inertial measurement unit for electronically stabilized event-based vision," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2014, pp. 2636–2639.

[94] R. Berner, "Highspeed USB2.0 AER interfaces," Master's thesis, ETH Zurich, Dept. of Electrical and Information Eng. (D-ITET), Zurich, Switzerland, 2006.

[95] G. Taverni, D. P. Moeys, C. Li, C. Cavaco, V. Motsnyi, D. S. S. Bello, and T. Delbruck, "Front and back illuminated Dynamic and Active Pixel Vision Sensors comparison," *IEEE Trans. Circuits Syst. II*, vol. 65, no. 5, pp. 677–681, 2018.

[96] D. B. Fasnacht and T. Delbruck, "Dichromatic spectral measurement circuit in vanilla CMOS," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2007, pp. 3091–3094.

[97] R. Berner, P. Lichtsteiner, and T. Delbruck, "Self-timed vertacolor dichromatic vision sensor for low power pattern detection," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2008, pp. 1032–1035.

[98] L. Farian, J. A. Leñero-Bardallo, and P. Häfliger, "A bio-inspired AER temporal tri-color differentiator pixel array," *IEEE Trans. Biomed. Circuits Syst.*, vol. 9, no. 5, pp. 686–698, Oct. 2015.

[99] R. B. Merrill, "Color separation in an active pixel cell imaging array using a triple-well structure," US Patent US5 965 875A, 1999.

[100] R. F. Lyon and P. M. Hubel, "Eyeing the camera: Into the next century," in *Proc. IS&T/SID 10th Color Imaging Conf.*, 2002, pp. 349–355.

[101] C. Li, C. Brandli, R. Berner, H. Liu, M. Yang, S.-C. Liu, and T. Delbruck, "An RGBW color VGA rolling and global shutter dynamic and active-pixel vision sensor," in *Int. Image Sensor Workshop (IISW)*, 2015.

[102] D. P. Moeys, C. Li, J. N. P. Martel, S. Bamford, L. Longinotti, V. Motsnyi, D. S. S. Bello, and T. Delbruck, "Color temporal contrast sensitivity in dynamic vision sensors," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2017, pp. 1–4.

[103] C. Scheerlinck, H. Rebecq, T. N. Stoffregen, N. Barnes, R. Mahony, and D. Scaramuzza, "CED: Color event camera dataset," in *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, 2019.

[104] A. Marcireau, S.-H. Ieng, C. Simon-Chane, and R. B. Benosman, "Event-based color segmentation with a high dynamic range sensor," *Front. Neurosci.*, vol. 12, 2018.

[105] C. Posch, D. Matolin, R. Wohlgenannt, T. Maier, and M. Litzenberger, "A microbolometer asynchronous Dynamic Vision Sensor for LWIR," *IEEE Sensors J.*, vol. 9, no. 6, pp. 654–664, Jun. 2009.

[106] H. Akolkar, S. Panzeri, and C. Bartolozzi, "Spike time based unsupervised learning of receptive fields for event-driven vision," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2015.

[107] J. A. Perez-Carrasco, B. Zhao, C. Serrano, B. Acha, T. Serrano-Gotarredona, S. Chen, and B. Linares-Barranco, "Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feed-forward ConvNets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2706–2719, Nov. 2013.

[108] P. O'Connor, D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer, "Real-time classification and sensor fusion with a spiking deep belief network," *Front. Neurosci.*, vol. 7, p. 178, 2013.

[109] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, and M. Pfeiffer, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 4, 2015, pp. 2933–2940.

[110] S. K. Esser, P. A. Merolla, J. V. Arthur, A. S. Cassidy, R. Appuswamy, A. Andreopoulos, D. J. Berg, J. L. McKinstry, T. Melano, D. R. Barch, C. di Nolfo, P. Datta, A. Amir, B. Taba, M. D. Flickner, and D. S. Modha, "Convolutional networks for fast, energy-efficient neuromorphic computing," *Proc. National Academy of Sciences*, vol. 113, no. 41, pp. 11 441–11 446, 2016.

[111] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Front. Neurosci.*, vol. 11, p. 682, 2017.

[112] S. B. Shrestha and G. Orchard, "SLAYER: Spike layer error reassignment in time," in *Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2018.

[113] J. Kogler, C. Sulzbachner, and W. Kubinger, "Bio-inspired stereo vision system with silicon retina imagers," in *Int. Conf. Comput. Vis. Syst. (ICVS)*, 2009, pp. 174–183.

[114] J. Kogler, C. Sulzbachner, M. Humenberger, and F. Eibensteiner, "Address-event based stereo vision with bio-inspired silicon retina imagers," in *Advances in Theory and Applications of Stereo Vision*. InTech, 2011, pp. 165–188.

[115] M. Liu and T. Delbruck, "Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors," in *British Mach. Vis. Conf. (BMVC)*, 2018.

[116] G. Gallego and D. Scaramuzza, "Accurate angular velocity estimation with an event camera," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 632–639, 2017.

[117] A. Z. Zhu, N. Atanasov, and K. Daniilidis, "Event-based feature tracking with probabilistic data association," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2017, pp. 4465–4470.

[118] E. Mueggler, G. Gallego, and D. Scaramuzza, "Continuous-time trajectory estimation for event-based vision sensors," in *Robotics: Science and Systems (RSS)*, 2015.

[119] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza, "Continuous-time visual-inertial odometry for event cameras," *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1425–1440, Dec. 2018.

[120] A. Z. Zhu, N. Atanasov, and K. Daniilidis, "Event-based visual inertial odometry," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2017, pp. 5816–5824.

[121] G. Gallego, M. Gehrig, and D. Scaramuzza, "Focus is all you need: Loss functions for event-based vision," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.

[122] Z. Ni, S.-H. Ieng, C. Posch, S. Régnier, and R. Benosman, "Visual tracking using neuromorphic asynchronous event-based cameras," *Neural Computation*, vol. 27, no. 4, pp. 925–953, 2015.

[123] E. Mueggler, C. Bartolozzi, and D. Scaramuzza, "Fast event-based corner detection," in *British Mach. Vis. Conf. (BMVC)*, 2017.

[124] I. Alzugaray and M. Chli, "Asynchronous corner detection and tracking for event cameras in real time," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3177–3184, Oct. 2018.

[125] D. P. Moeys, F. Corradi, E. Kerr, P. Vance, G. Das, D. Neil, D. Kerr, and T. Delbruck, "Steering a predator robot using a mixed frame/event-driven convolutional neural network," in *Int. Conf. Event-Based Control, Comm. Signal Proc. (EBCCSP)*, 2016.

[126] I.-A. Lungu, F. Corradi, and T. Delbruck, "Live demonstration: Convolutional neural network driven by dynamic vision sensor playing RoShamBo," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2017.

[127] J. Binas, D. Neil, S.-C. Liu, and T. Delbruck, "DDD17: End-to-end DAVIS driving dataset," in *ICML Workshop on Machine Learning for Autonomous Vehicles*, 2017.

[128] A. I. Maqueda, A. Loquercio, G. Gallego, N. García, and D. Scaramuzza, "Event-based vision meets deep learning on steering prediction for self-driving cars," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2018, pp. 5419–5427.

[129] C. Ye, A. Mitrokhin, C. Parameshwara, C. Fermüller, J. A. Yorke, and Y. Aloimonos, "Unsupervised learning of dense optical flow and depth from sparse event data," *arXiv e-prints*, 2018, 1809.08625.

[130] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.

[131] A. Nguyen, T. Do, D. G. Caldwell, and N. G. Tsagarakis, "Real-time pose estimation for event cameras with stacked spatial LSTM networks," *arXiv e-prints*, 2017, 1708.09011.

[132] A. Mitrokhin, C. Ye, C. Fermuller, Y. Aloimonos, and T. Delbruck, "EV-IMO: Motion segmentation dataset and learning pipeline for event cameras," *arXiv preprint arXiv:1903.07520*, 2019.

[133] D. Gehrig, A. Loquercio, K. G. Derpanis, and D. Scaramuzza, "End-to-end learning of representations for asynchronous event-based data," *arXiv e-prints*, 2019.

[134] Y. Sekikawa, K. Hara, and H. Saito, "EventNet: Asynchronous recursive event processing," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.

[135] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Conf. Neural Inf. Process. Syst. (NIPS)*, 2017.

[136] L. A. Camunas-Mesa, T. Serrano-Gotarredona, and B. Linares-Barranco, "Event-driven sensing and processing for high-speed robotic vision," in *IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, 2014.

[137] G. Orchard, R. Benosman, R. Etienne-Cummings, and N. V. Thakor, "A spiking neural network architecture for visual motion estimation," in *IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, 2013, pp. 298–301.

[138] S. Tschechne, R. Sailer, and H. Neumann, "Bio-inspired optic flow from event-based neuromorphic sensor input," in *Artificial Neural Networks in Pattern Recognition (ANNPR)*, 2014, pp. 171–182.

[139] E. Chicca, P. Lichtsteiner, T. Delbruck, G. Indiveri, and R. Douglas, "Modeling orientation selectivity using a neuromorphic multi-chip system," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2006, pp. 1235–1238.

[140] R. L. D. Valois, N. P. Cottaris, L. E. Mahon, S. D. Elfar, and J. Wilson, "Spatial and temporal receptive fields of geniculate and cortical cells and directional selectivity," *Vision Research*, vol. 40, no. 27, pp. 3685–3702, 2000.

[141] F. Rea, G. Metta, and C. Bartolozzi, "Event-driven visual attention for the humanoid robot iCub," *Front. Neurosci.*, vol. 7, 2013.

[142] L. Itti and C. Koch, "Computational modelling of visual attention," *Nature Reviews Neuroscience*, vol. 2, no. 3, pp. 194–203, Mar. 2001.

[143] D. Marr and T. Poggio, "Cooperative computation of stereo disparity," *Science*, vol. 194, no. 4262, pp. 283–287, 1976.

[144] M. Mahowald, *The Silicon Retina*. Boston, MA: Springer US, 1994, pp. 4–65.

[145] M. Osswald, S.-H. Ieng, R. Benosman, and G. Indiveri, "A spiking neural network model of 3D perception for event-based neuromorphic stereo vision systems," *Scientific Reports*, vol. 7, no. 1, Jan. 2017.

[146] G. Dikov, M. Firouzi, F. Röhrbein, J. Conradt, and C. Richter, "Spiking cooperative stereo-matching at 2ms latency with neuromorphic hardware," in *Conf. Biomimetic and Biohybrid Systems*, 2017, pp. 119–137.

[147] E. Piatkowska, J. Kogler, N. Belbachir, and M. Gelautz, "Improved cooperative stereo matching for dynamic vision sensors with ground truth evaluation," in *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, 2017.

[148] V. Vasco, A. Glover, Y. Tirupachuri, F. Solari, M. Chessa, and C. Bartolozzi, "Vergence control with a neuromorphic iCub," in *IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, 2016.

[149] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neuroscience*, vol. 2, no. 11, pp. 1019–1025, Nov. 1999.

[150] H. Akolkar, C. Meyer, X. Clady, O. Marre, C. Bartolozzi, S. Panzeri, and R. Benosman, "What can neuromorphic event-driven precise timing add to spike-based pattern recognition?" *Neural Computation*, vol. 27, no. 3, pp. 561–593, Mar. 2015.

[151] L. A. Camunas-Mesa, T. Serrano-Gotarredona, S. H. Ieng, R. B. Benosman, and B. Linares-Barranco, "On the use of orientation filters for 3D reconstruction in event-driven stereo vision," *Front. Neurosci.*, vol. 8, p. 48, 2014.

[152] M. B. Milde, D. Neil, A. Aimar, T. Delbrück, and G. Indiveri, "ADaPTION: Toolbox and benchmark for training convolutional neural networks with reduced numerical precision weights and activation," *arXiv e-prints*, Nov. 2017.

[153] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Front. Neurosci.*, vol. 10, p. 508, 2016.

[154] E. Stromatias, M. Soto, T. Serrano-Gotarredona, and B. Linares-Barranco, "An event-driven classifier for spiking neural networks fed with synthetic or dynamic vision sensor data," *Front. Neurosci.*, vol. 11, Jun. 2017.

[155] E. Neftci, C. Augustine, S. Paul, and G. Detorakis, "Neuromorphic deep learning machines," *arXiv e-prints*, 2016, 1612.05596.

[156] M. B. Milde, O. J. N. Bertrand, H. Ramachandran, M. Egelhaaf, and E. Chicca, "Spiking elementary motion detector in neuromorphic systems," *Neural Computation*, vol. 30, no. 9, pp. 2384–2417, Sep. 2018.

[157] H. Blum, A. Dietmller, M. Milde, J. Conradt, G. Indiveri, and Y. Sandamirskaya, "A neuromorphic controller for a robotic vehicle equipped with a dynamic vision sensor," in *Robotics: Science and Systems (RSS)*, 2017.

[158] L. Salt, D. Howard, G. Indiveri, and Y. Sandamirskaya, "Differential evolution and bayesian optimisation for hyper-parameter selection in mixed-signal neuromorphic circuits applied to UAV obstacle avoidance," *arXiv e-prints*, 2017, 1704.04853.

[159] S.-H. Ieng, C. Posch, and R. Benosman, "Asynchronous neuromorphic event-driven image filtering," *Proc. IEEE*, vol. 102, no. 10, pp. 1485–1499, Oct. 2014.

[160] D. Gehrig, H. Rebecq, G. Gallego, and D. Scaramuzza, "Asynchronous, photometric feature tracking using events and frames," in *Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 766–781.

[161] M. Litzenberger, C. Posch, D. Bauer, A. N. Belbachir, P. Schön, B. Kohn, and H. Garn, "Embedded vision system for real-time object tracking using an asynchronous transient vision sensor," in *Digital Signal Processing Workshop*, 2006, pp. 173–178.

[162] Z. Ni, C. Pacoret, R. Benosman, S.-H. Ieng, and S. Régnier, "Asynchronous event-based high speed vision for microparticle tracking," *J. Microscopy*, vol. 245, no. 3, pp. 236–244, 2012.

[163] Z. Ni, A. Bolopion, J. Agnus, R. Benosman, and S. Régnier, "Asynchronous event-based visual shape tracking for stable haptic feedback in microrobotics," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1081–1089, 2012.

[164] X. Lagorce, C. Meyer, S.-H. Ieng, D. Filliat, and R. Benosman, "Asynchronous event-based multikernel algorithm for high-speed visual features tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1710–1720, Aug. 2015.

[165] D. R. Valeiras, X. Lagorce, X. Clady, C. Bartolozzi, S.-H. Ieng, and R. Benosman, "An asynchronous neuromorphic event-driven visual part-based shape tracking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3045–3059, Dec. 2015.

[166] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Trans. Comput.*, vol. C-22, no. 1, pp. 67–92, Jan. 1973.

[167] D. Tedaldi, G. Gallego, E. Mueggler, and D. Scaramuzza, "Feature detection and tracking with the dynamic and active-pixel vision sensor (DAVIS)," in *Int. Conf. Event-Based Control, Comm. Signal Proc. (EBCCSP)*, 2016.

[168] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Fourth Alvey Vision Conf.*, vol. 15, 1988, pp. 147–151.

[169] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Int. Joint Conf. Artificial Intell. (IJCAI)*, 1981, pp. 674–679.

[170] X. Clady, S.-H. Ieng, and R. Benosman, "Asynchronous event-based corner detection and matching," *Neural Netw.*, vol. 66, pp. 91–106, 2015.

[171] E. Mueggler, C. Forster, N. Baumli, G. Gallego, and D. Scaramuzza, "Lifetime estimation of events from dynamic vision sensors," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2015, pp. 4874–4881.

[172] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Eur. Conf. Comput. Vis. (ECCV)*, 2006, pp. 430–443.

[173] V. Vasco, A. Glover, and C. Bartolozzi, "Fast event-based Harris corner detection exploiting the advantages of event-driven cameras," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2016.

[174] J. Manderscheid, A. Sironi, N. Bourdis, D. Migliore, and V. Lepetit, "Speed invariant time surface for learning to detect corner points with event-based cameras," in *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.

[175] V. Vasco, A. Glover, E. Mueggler, D. Scaramuzza, L. Natale, and C. Bartolozzi, "Independent motion detection with event-driven cameras," in *IEEE Int. Conf. Adv. Robot. (ICAR)*, 2017.

[176] Y. Hu, H. Liu, M. Pfeiffer, and T. Delbruck, "DVS benchmark datasets for object tracking, action recognition, and object recognition," *Front. Neurosci.*, vol. 10, p. 405, 2016.

[177] F. Barranco, C. Fermuller, and Y. Aloimonos, "Contour motion estimation for asynchronous event-driven cameras," *Proc. IEEE*, vol. 102, no. 10, pp. 1537–1556, Oct. 2014.

[178] ——, "Bio-inspired motion estimation with event-driven sensors," in *Int. Work-Conf. Artificial Neural Netw. (IWANN), Advances in Comput. Intell.*, 2015, pp. 309–321.

[179] T. Brosch, S. Tschechne, and H. Neumann, "On event-based optical flow detection," *Front. Neurosci.*, vol. 9, Apr. 2015.

[180] M. Liu and T. Delbruck, "Block-matching optical flow for dynamic vision sensors: Algorithm and FPGA implementation," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2017.

[181] T. Stoffregen and L. Kleeman, "Simultaneous optical flow and segmentation (SOFAS) using Dynamic Vision Sensor," in *Australasian Conf. Robot. Autom. (ACRA)*, 2017.

[182] G. Haessig, A. Cassidy, R. Alvarez-Icaza, R. Benosman, and G. Orchard, "Spiking optical flow for event-based sensors using IBM's truenorth neurosynaptic system," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 4, pp. 860–870, Aug. 2018.

[183] A. Z. Zhu, D. Thakur, T. Ozaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The multivehicle stereo event camera dataset: An event camera dataset for 3D perception," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2032–2039, Jul. 2018.

[184] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003, 2nd Edition.

[185] S. Schraml, A. N. Belbachir, N. Milosevic, and P. Schön, "Dynamic stereo vision system for real-time tracking," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2010, pp. 1409–1412.

[186] J. Kogler, M. Humenberger, and C. Sulzbachner, "Event-based stereo matching approaches for frameless address event stereo data," in *Int. Symp. Adv. Vis. Comput. (ISVC)*, 2011, pp. 674–685.

[187] J. Lee, T. Delbruck, P. K. J. Park, M. Pfeiffer, C.-W. Shin, H. Ryu, and B. C. Kang, "Live demonstration: Gesture-based remote control using stereo pair of dynamic vision sensors," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2012.

[188] R. Benosman, S.-H. Ieng, P. Rogister, and C. Posch, "Asynchronous event-based Hebbian epipolar geometry," *IEEE Trans. Neural Netw.*, vol. 22, no. 11, pp. 1723–1734, 2011.

[189] J. Carneiro, S.-H. Ieng, C. Posch, and R. Benosman, "Event-based 3D reconstruction from neuromorphic retinas," *Neural Netw.*, vol. 45, pp. 27–38, 2013.

[190] D. Zou, P. Guo, Q. Wang, X. Wang, G. Shao, F. Shi, J. Li, and P.-K. J. Park, "Context-aware event-driven stereo matching," in *IEEE Int. Conf. Image Process. (ICIP)*, 2016, pp. 1076–1080.

[191] D. Zou, F. Shi, W. Liu, J. Li, Q. Wang, P.-K. J. Park, C.-W. Shi, Y. J. Roh, and H. E. Ryu, "Robust dense depth map estimation from sparse DVS stereos," in *British Mach. Vis. Conf. (BMVC)*, 2017.

[192] M. Firouzi and J. Conradt, "Asynchronous event-based cooperative stereo matching using neuromorphic silicon retinas," *Neural Proc. Lett.*, vol. 43, no. 2, pp. 311–326, 2015.

[193] Z. Xie, J. Zhang, and P. Wang, "Event-based stereo matching using semiglobal matching," *Int. J. Advanced Robotic Systems*, vol. 15, no. 1, 2018.

[194] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[195] A. Z. Zhu, Y. Chen, and K. Daniilidis, "Realtime time synchronized event-based stereo," in *Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 438–452.

[196] H. Rebecq, G. Gallego, and D. Scaramuzza, "EMVS: Event-based multi-view stereo," in *British Mach. Vis. Conf. (BMVC)*, 2016.

[197] R. Szeliski, *Computer Vision: Algorithms and Applications*, ser. Texts in Computer Science. Springer, 2010.

[198] S. Schraml, A. N. Belbachir, and H. Bischof, "An event-driven stereo system for real-time 3-D 360 panoramic vision," *IEEE Trans. Ind. Electron.*, vol. 63, no. 1, pp. 418–428, Jan. 2016.

[199] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. D. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016.

[200] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Int. Conf. Comput. Vis. (ICCV)*, 2011, pp. 2320–2327.

[201] Y. Zhou, G. Gallego, H. Rebecq, L. Kneip, H. Li, and D. Scaramuzza, "Semi-dense 3D reconstruction with a stereo event camera," in *Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 242–258.

[202] C. Brandli, T. Mantel, M. Hutter, M. Höpflinger, R. Berner, R. Siegwart, and T. Delbruck, "Adaptive pulsed laser line extraction for terrain reconstruction using a dynamic vision sensor," *Front. Neurosci.*, vol. 7, p. 275, 2014.

[203] J. N. P. Martel, J. Müller, J. Conradt, and Y. Sandamirskaya, "An active approach to solving the stereo matching problem using event-based sensors," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2018, pp. 1–5.

[204] C. Reinbacher, G. Munda, and T. Pock, "Real-time panoramic tracking for event cameras," in *IEEE Int. Conf. Comput. Photography (ICCP)*, 2017, pp. 1–9.

[205] D. Weikersdorfer, D. B. Adrian, D. Cremers, and J. Conradt, "Event-based 3D SLAM with a depth-augmented dynamic vision sensor," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014, pp. 359–364.

[206] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM," *Int. J. Robot. Research*, vol. 36, no. 2, pp. 142–149, 2017.

[207] M. Milford, H. Kim, S. Leutenegger, and A. Davison, "Towards visual SLAM with event-based cameras," in *The Problem of Mobile Sensors Workshop in conjunction with RSS*, 2015.

[208] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, Apr. 2007, pp. 3565–3572.

[209] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial SLAM using nonlinear optimization," *Int. J. Robot. Research*, 2015.

[210] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," *IEEE Trans. Robot.*, vol. 33, no. 1, pp. 1–21, 2017.

[211] A. Patron-Perez, S. Lovegrove, and G. Sibley, "A spline-based trajectory representation for sensor fusion and rolling shutter cameras," *Int. J. Comput. Vis.*, vol. 113, no. 3, pp. 208–219, 2015.

[212] L. von Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2018.

[213] S. Barua, Y. Miyatani, and A. Veeraraghavan, "Direct face detection and video reconstruction from event cameras," in *IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, 2016, pp. 1–9.

[214] A. N. Belbachir, S. Schraml, M. Mayerhofer, and M. Hofstaetter, "A novel HDR depth camera for real-time 3D 360-degree panoramic vision," in *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*, 2014.

[215] G. Munda, C. Reinbacher, and T. Pock, "Real-time intensity-image reconstruction for event cameras using manifold regularisation," *Int. J. Comput. Vis.*, vol. 126, no. 12, pp. 1381–1393, Jul. 2018.

[216] S. Mostafavi I., L. Wang, Y.-S. Ho, and K.-J. Y. Yoon, "Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks," *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, 2019.

[217] A. Mitrokhin, C. Fermuller, C. Parameshwara, and Y. Aloimonos, "Event-based moving object detection and tracking," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2018.

[218] T. Stoffregen, G. Gallego, T. Drummond, L. Kleeman, and D. Scaramuzza, "Event-based motion segmentation by motion compensation," *arXiv preprint arXiv:1904.01293*, 2019.

[219] E. Neftci, "Data and power efficient intelligence with neuromorphic learning machines," *iScience*, 2018.

[220] G. Orchard, A. Jayawant, G. K. Cohen, and N. Thakor, "Converting static image datasets to spiking neuromorphic datasets using saccades," *Front. Neurosci.*, vol. 9, p. 437, 2015.

[221] D. Neil and S.-C. Liu, "Effective sensor fusion with event-based sensors and deep network architectures," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2016, pp. 2282–2285.

[222] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Front. Neurosci.*, vol. 12, 2018.

[223] A. Yousefzadeh, G. Orchard, T. Serrano-Gotarredona, and B. Linares-Barranco, "Active perception with dynamic vision sensors. minimum saccades with optimum recognition," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 4, pp. 927–939, Aug. 2018.

[224] X. Clady, J.-M. Maro, S. Barré, and R. B. Benosman, "A motion-based feature for event-based pattern recognition," *Front. Neurosci.*, vol. 10, Jan. 2017.

[225] C. A. Sims, "Implications of rational inattention," *Journal of Monetary Economics*, vol. 50, pp. 665–690, 04 2003.

[226] M. Miskowicz, *Event-Based Control and Signal Processing*, ser. Embedded Systems. CRC Press, 2018.

[227] P. Tabuada, "An introduction to event-triggered and self-triggered control," in *IEEE Conf. Decision Control (CDC)*, 2012.

[228] K. J. Aström, *Event Based Control*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 127–147.

[229] B. Wang and M. Fu, "Comparison of periodic and event-based sampling for linear state estimation," *IFAC Proc. Volumes (IFAC-PapersOnline)*, vol. 19, pp. 5508–5513, 2014.

[230] A. Censi, E. Mueller, E. Frazzoli, and S. Soatto, "A power-performance approach to comparing sensor families, with application to comparing neuromorphic to traditional vision sensors," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2015.

[231] E. Mueller, A. Censi, and E. Frazzoli, "Low-latency heading feedback control with neuromorphic vision sensors using efficient approximated incremental inference," in *IEEE Conf. Decision Control (CDC)*, 2015.

[232] P. Singh, S. Z. Yong, J. Gregoire, A. Censi, and E. Frazzoli, "Stabilization of linear continuous-time systems using neuromorphic vision sensors," in *IEEE Conf. Decision Control (CDC)*, 2016.

[233] A. Censi, "Efficient neuromorphic optomotor heading regulation," in *Proceedings of the American Control Conference*, vol. 2015-July, 2015, pp. 3854–3861.

[234] E. Mueller, A. Censi, and E. Frazzoli, "Efficient high speed signal estimation with neuromorphic vision sensors," in *Int. Conf. Event-Based Control, Comm. Signal Proc. (EBCCSP)*, 2015.

[235] E. Mueller, "Motion Planning for Autonomous Vehicles in Unstructured Environments Using Dynamic Vision Sensors," pp. 1–38, may 2014.

[236] D. Kahneman, *Thinking, fast and slow*. Farrar, Straus and Giroux, 2011.

[237] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown, "Overview of the SpiNNaker system architecture," *IEEE Trans. Comput.*, vol. 62, no. 12, pp. 2454–2467, 2013.

[238] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, G.-J. Nam, B. Taba, M. Beakes, B. Brezzo, J. B. Kuang, R. Manohar, W. P. Risk, B. Jackson, and D. S. Modha, "TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 10, pp. 1537–1557, 2015.

[239] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[240] S. Moradi, N. Qiao, F. Stefanini, and G. Indiveri, "A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs)," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 106–122, 2018.

[241] A. S. Neckar, "Braindrop: a mixed signal neuromorphic architecture with a dynamical systems-based programming model," Ph.D. dissertation, Stanford University, Stanford, CA, Jun. 2018.

[242] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[243] P. Blouw, X. Choo, E. Hunsberger, and C. Eliasmith, "Benchmarking Keyword Spotting Efficiency on Neuromorphic Hardware," *arXiv e-prints*, p. arXiv:1812.01739, Dec. 2018.

[244] N. Waniek, J. Biedermann, and J. Conradt, "Cooperative SLAM on small mobile robots," in *IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, 2015.

[245] B. J. P. Hordijk, K. Y. Scheper, and G. C. D. Croon, "Vertical landing for micro air vehicles using event-based optical flow," *J. Field Robot.*, vol. 35, no. 1, pp. 69–90, Jan. 2017.

[246] https://github.com/uzh-rpg/event-based_vision_resources, 2017.

[247] https://github.com/SensorsINI/jaer, 2007.

[248] T. Delbruck, M. Pfeiffer, R. Juston, G. Orchard, E. Müggler, A. Linares-Barranco, and M. W. Tilden, "Human vs. computer slot car racing using an event and frame-based DAVIS vision sensor," in *IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2015, pp. 2409–2412.

[249] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop Open Source Softw.*, 2009.

[250] A. Glover, V. Vasco, M. Iacono, and C. Bartolozzi, "The event-driven software library for YARP-with algorithms and iCub applications," *Front. Robotics and AI*, vol. 4, p. 73, 2018.

[251] F. Barranco, C. Fermuller, Y. Aloimonos, and T. Delbruck, "A dataset for visual navigation with neuromorphic methods," *Front. Neurosci.*, vol. 10, p. 49, 2016.

[252] J. Delmerico, T. Cieslewski, H. Rebecq, M. Faessler, and D. Scaramuzza, "Are we ready for autonomous drone racing? the UZH-FPV drone racing dataset," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019.

[253] C. Tan, S. Lallee, and G. Orchard, "Benchmarking neuromorphic vision: lessons learnt from computer vision," *Front. Neurosci.*, vol. 9, p. 374, 2015.

[254] S. J. Carey, A. Lopich, D. R. Barr, B. Wang, and P. Dudek, "A 100,000 fps vision sensor with embedded 535 GOPS/W 256x256 SIMD processor array," in *VLSI Circuits Symp.*, 2013.