10

Event-based SLAM

Guillermo Gallego, Javier Hidalgo-Carrió, and Davide Scaramuzza

An inquisitive reader would notice that SLAM is paramount in applications that involve interpretation of spatial relationships and interaction with the surroundings to solve complex real-world problems. SLAM's primary sensors are critical for the system's success and adaptability. Visual SLAM is the most extended category of all because cameras are broadly available (affordable) and produce an intuitive and informative signal that allows us to sense the world in a wide range of scenarios (e.g., yielding lightweight systems that do not require additional infrastructure like GNSS). Despite the progress so far, state-of-the-art artificial intelligence systems are not as effective (robust and efficient) in real-world tasks as their biological counterparts. Standard cameras sense the world at a fixed frame rate that is independent of the scene dynamics. Thus, they become blind in the time between frames, introduce latency, potentially lose tracking, and produce large amounts of redundant data if nothing moves in the scene. This chapter pursues the visionary challenge of understanding and building visual SLAM systems that are fast (not limited by a frame rate), low-power, and robust to broad illumination conditions by leveraging the bioinspired technology of silicon retinas or "event cameras", which overcome several of the limitations of standard cameras. See Fig. 10.1.

10.1 Sensor Description

10.1.1 Working principle

In contrast to traditional cameras, which acquire full images at a rate given by an external clock (e.g., 30 Hz), the pixels of event cameras like the Dynamic Vision Sensor (DVS) [668, 369] operate independently from each other, responding to brightness changes in the scene asynchronously, as they occur (Figure 10.2b). These pixelwise changes are due to scene illumination (e.g., flickering lights) and/or to the relative motion of the camera and the scene (including moving objects). Hence, the output of an event camera is a sequence of digital "events" (or "spikes"), where each event represents a change of brightness (logarithmic intensity). This encoding is inspired by the spiking nature of biological visual pathways (Figure 10.2a).

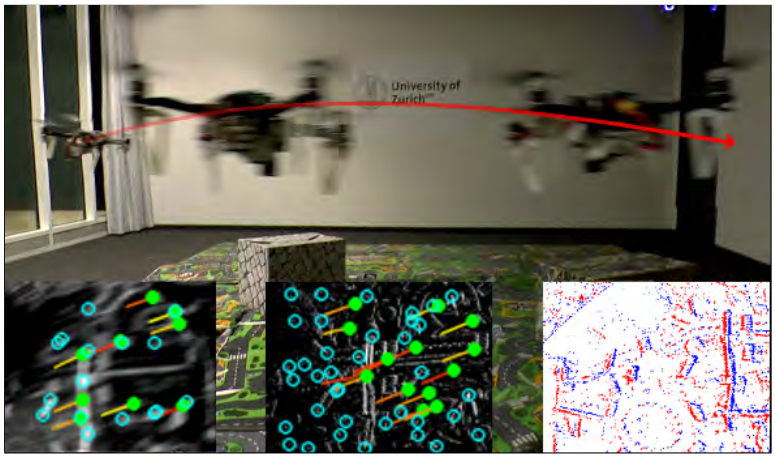


Figure 10.1 Drone with a downlooking DAVIS camera [117] (240×180 px) performing an autonomous flight using a visual-inertial odometry (VIO) algorithm [949] for state estimation. The high speed and high dynamic range of the event camera data are leveraged to operate in difficult illumination conditions. The insets show features (i.e., keypoints) detected and tracked in grayscale frames (left, motion-blurred) and in motion-compensated images of warped events (middle, sharp). The event data (in red/blue according to polarity) clearly respond to the scene contours. The same VIO algorithm [949] is also demonstrated on high-speed scenarios, such as an event camera spinning tied to a rope. Image from [369] (©2020 IEEE).

Specifically, each pixel memorizes the logarithmic intensity L each time it sends an event, and continuously monitors for a change ΔL of sufficient magnitude from this memorized value (Figure 10.2). When the change reaches a threshold C,

$$\Delta L \doteq L(\mathbf{x}_k, t_k) - L(\mathbf{x}_k, t_k - \Delta t_k) = p_k C, \tag{10.1}$$

the camera sends an event, $e_k \doteq (\mathbf{x}_k, t_k, p_k)$, which is transmitted from the chip with the x, y pixel location $\mathbf{x}_k$, the time t_k, and the 1-bit polarity $p_k \in \{+1, -1\}$ of the change (i.e., brightness increase or decrease). Δt_k is the time elapsed since the previous event at the same pixel.

Event cameras are data-driven sensors: their output depends on the amount of motion or illumination change in the scene. The faster the motion, the more events per second are produced because each pixel adapts its sampling rate to the rate of change of the intensity signal that it monitors.

Bio-inspiration: The transient pathway. Event cameras are inspired by the operation of biological visual pathways, which are the information processing routes in animals and humans. Following the two-stream hypothesis, the dorsal stream (also called "transient" or "where" pathway) is dedicated to processing dynamic visual information (e.g., motion in the scene), whereas the ventral stream (called "sustained" or "what" pathway) is dedicated to object and visual identification and

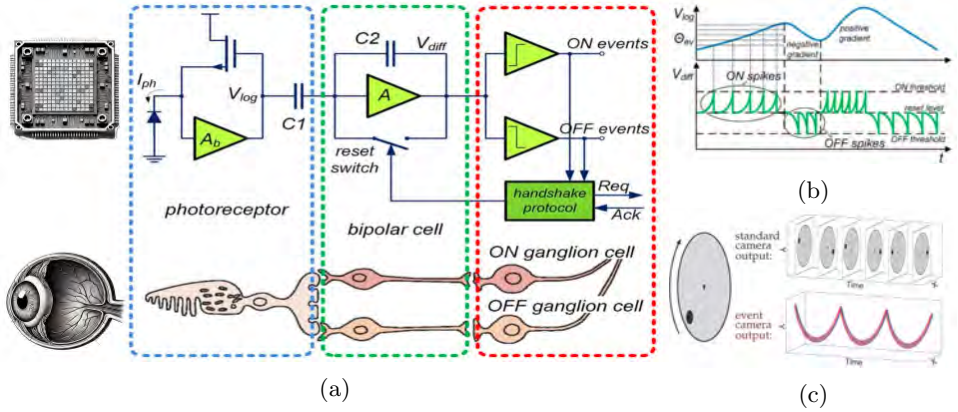


Figure 10.2 Working principle of an event camera (e.g., DVS): (a) Three-layer model of a human retina and corresponding DVS pixel circuit; (b) Schematic of the operation of a DVS pixel, converting light into events (spikes), with the colors of the signals matching those of the layers in (a); (c) Comparison of the response of a standard camera and an event camera to a visual stimulus consisting of a black dot on a rotating disk. An event camera transmits the brightness changes continuously, forming a spiral of events in space-time. Red color: positive events (ON spikes), blue color: negative events (OFF spikes). Image adapted from [891]. TODO: Reuse permission for (a) and (b)

recognition. The DVS [668] corresponds to the part of the transient pathway from the photoreceptors up to the ganglion cells, adopting a simplified 3-layer pixel design that balances biological fidelity and circuitry stability (Figure 10.2). The three layers realize the functions of light conversion, delta-modulation and comparison, respectively. Cameras like the Asynchronous time-based image sensor (ATIS) [890] or the Dynamic and Active-Pixel Vision Sensor (DAVIS) [117] model both visual pathways, and therefore output two types of signals: DVS events and grayscale information (e.g., images). More details of the main event camera types are provided in [891, 369].

10.1.2 Advantages of Event Cameras

The sensing principle of event cameras is radically different from that of standard (exposure-based) cameras that have dominated computer and robot vision for the last seven decades, and it offers numerous advantages:

High Temporal Resolution: events are detected and timestamped with microsecond resolution, which enables capturing very fast motions without suffering from motion blur typical of frame-based cameras. Events are produced almost continuously in time, thus avoiding blind times that can cause large inter-image displacements and ruin data association in standard cameras.

Low Latency: each pixel works independently, without waiting for a global exposure time, thus events are transmitted as soon as a brightness change is detected, with submillisecond latency.

Low Power and Bandwith: events represent non-redundant temporal data, hence power is purposely spent. Bandwidth is also reduced (compared to a traditional camera operating at the same rate). At the die level, cameras consume less than 10 mW, allowing embedded systems to consume 100 mW or less [38].

High Dynamic Range (HDR): the range of light values that event cameras can sense is very high (typically >120 dB vs. 60 dB of standard cameras), enabling them to sense very dark (moonlight) and very bright (daylight) regions, simultaneously. Hence, they do not suffer from under/over-exposure typical of frame-based cameras. This property is due two facts: each pixel works independently and converts light to voltage in logarithmic scale.

10.1.3 Current Devices and Trends.

Which event camera should I buy or use to solve my SLAM problem? We often get asked this question by people entering this emerging field. The characteristics of event cameras are often compared via tables [369, Tab. 1], [168, Tabs. 1–2]. Although multiple event camera designs exist, most of them are laboratory prototypes. Only a few make it into commercialized devices that enable the exploration of novel solutions to classical as well as new problems, such as event-based SLAM. Among the devices commercialized by the main manufacturers (SONY, Samsung, iniVation / SynSense, Prophesee, Omnivision), some trends are worth mentioning:

Pixel size: following the megapixel race of traditional cameras and pressure from industry requirements, the pixel pitch (i.e., size) has considerably decreased, from 40 µm (DVS128 [668]) to less than 5 µm [334]. DVS pixels carry out more operations (modulation, comparison, etc.) than their traditional counterparts; hence, they require more transistors, which are more difficult to pack in the same sensor area. To maximize the area of the pixels exposed to light (that is, the fill factor) and reduce the gap between the photoreceptive parts of the pixels, stacked technology and backside illumination have been adopted [334].

Grayscale output: early devices such as the DAVIS or ATIS concurrently output grayscale data (e.g., images [117]), which is especially useful in applications with stationary cameras (albeit this is not the usual scenario in SLAM). Newer models such as HD event cameras [334] discontinued the grayscale output in favor of more area for the event output, driven by the megapixel race.

Color is not essential in many motion-related tasks, and therefore only a couple of event camera models offer color filters to detect changes in respective color channels (red, green and blue – RGB) [776].

Inertial data: some cameras also provide data from an inertial measurement unit (IMU) integrated in the same device. IMUs are valuable complementary proprio-

ceptive sensors to cameras, enabling visual-inertial odometry (VIO) (sensor fusion), yielding higher robustness and accuracy than single-sensor systems.

It is unrealistic to think that high-spatial–resolution event cameras are per se better than low-resolution ones. While capturing fine spatial details is important, noise and bandwidth also play an important role in the target application requirements. In SLAM and related tasks, where event cameras may move fast and/or over high-textured scenes, HD (1 Megapixel) event cameras can produce hundreds of millions of events per second. This poses problems, such as saturation of the output transmission bus of the camera), and high processing demands; currently there is no algorithm-and-hardware combination that can process such event rate in real time (without resorting to array-like conversion and/or sub/downsampling). New hybrid sensors, such as [1233], with lower spatial resolution for events than for intensity output, or foveated sensors [325], mimicking biological vision to decrease bandwidth), are being developed; they may provide alternative solutions to the above issue. In SLAM, a lower pixel resolution (e.g., QVGA) is preferred for algorithm prototyping and for real-time operation on computationally-constrained robots. Often the choice of field of view (optics) is as important as the pixel count.

10.2 Challenges and Applications

Event cameras represent a revolutionary technology in visual data acquisition. Hence, they pose the challenge of designing novel methods (algorithms and hardware) to process the acquired data and extract valuable information from it, unlocking the advantages of the sensor. In particular, the main challenges are:

Dealing with the space-time output: The output of event cameras is fundamentally different from that of standard cameras: events are asynchronous and spatially sparse, whereas images are synchronous and dense. Hence, visual SLAM algorithms designed for image sequences are not directly applicable to event data.

Dealing with motion-dependent data: Unlike images, each event contains binary (increase/decrease) brightness change information that depends not only on the scene texture, but also on the relative motion between the scene and the camera.

Dealing with noise and dynamic effects: Event cameras are noisy because of the inherent photon shot noise, transistor circuit noise, their dependency on the amount of incident light, non-idealities and low-power (sub-threshold) operation.

These challenges call for new approaches that rethink the space-time, photometric and stochastic nature of event data. In the context of SLAM, this poses questions such as: What is the best way to extract information from the events for pose or depth estimation? What map and camera trajectory representations shall be used that take into account the quasi-continuous temporal granularity and sparse nature of event data? How to establish correspondences (data association) under motion-dependent data? How to model the problem (and its solution) without introducing the typical bottlenecks of frame-based technology?

The above questions have been driving the research on event-based SLAM (Fig. 10.3). This topic has evolved both on its own and in conjunction with other tasks, i.e., research on event-based SLAM has fostered research on other event-based tasks. For example, the synergy between SLAM and image reconstruction (the task of recovering absolute intensity from events) has been leveraged as early as the first works [237, 578] (rotational-motion SLAM) and [579] (6-DoF SLAM). Event-based SLAM and optical flow estimation have been treated together in [237, 1238, 1016, 593].

10.3 Methodology Overview

Event-based SLAM methods can be broadly categorized in two, depending on how many events are processed simultaneously: (*i*) methods that operate on an *event-by-event basis*, where the state of the system (e.g., scene map and camera trajectory) can change upon the arrival of a single event, thus achieving minimum latency, and (*ii*) methods that operate on *groups / batches / slices / packets of events*, which introduce some latency. A key design choice in the latter category is how to select the size of the packet, for which many solutions have been proposed (e.g., fixed number of events, fixed temporal duration, and hybrid criteria).

Orthogonally, depending on how events are processed, model-based approaches and data-driven (i.e., machine learning) approaches can be distinguished. Mimicking the categorization in frame-based SLAM, event-based SLAM methods can be classified into *indirect* methods (feature-based, using event corners, lines, normal flow, etc.) and *direct* (using all events). This categorization is related to the type of objective or loss function used: geometric- vs. photometric-based (e.g., a function of the event polarity or the event rate/activity), and also to the overall philosophy: indirect methods typically have two steps (a feature extraction step, which "converts" events into geometric primitives, followed by a geometric SLAM pipeline), whereas direct methods typically comprise a single step that maps event data into motion and scenen parameters. In the latter, the event generation model (10.1) (or its linearized version [369]) is a cornerstone for designing estimation methods. Handling data association between events is a central problem in event-based vision, and SLAM in particular. Due to the high temporal resolution of event cameras, data association is typically handled by temporal and spatial vicinity; both hard-association and soft-association strategies have been explored.

Each of the above categories has advantages and disadvantages. The problem of solving SLAM with event cameras is challenging, and has been historically tackled with increasing complexity along several axes: the number of unknowns (degrees of freedom – DoFs), the type of motion (from rotational or 2D scenarios) to 6-DoF motion, the scene complexity (texture) and its motion (static vs. dynamic – independent moving objects – IMOs). Event-based SLAM is not an isolated problem; as mentioned in Sec. 10.2, it has connections with other problems (optical flow, tracking, segmentation, etc.), in stronger or weaker form depending on the

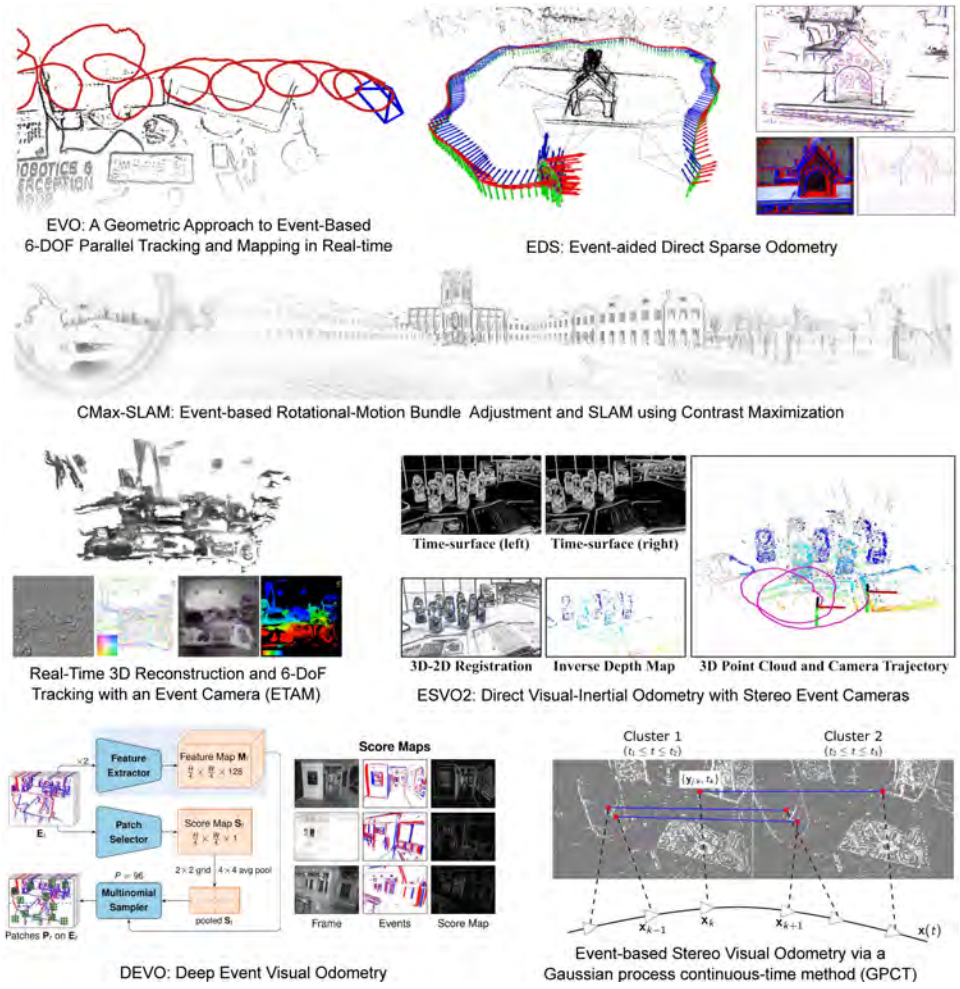


Figure 10.3 Event-based SLAM is actively being investigated, with systems that explore a large variety of approaches, including classical methods and more recent deep learning solutions. Since events are triggered by moving edges on the image plane, it is natural to recover scene maps in the form of edges (e.g., sparse or semi-dense 3D maps). Images adapted from EVO [921] (©2017 IEEE), EDS [469] (©2022 IEEE), CMax-SLAM [423] (©2024 IEEE), Kim et al. [579] (TODO: Reuse permission), ESVO2 [818] (©2025 IEEE), DEVO [593] (TODO: Reuse permission) and Wang et al. [1154] (TODO: Reuse permission).

assumptions or scenario considered. In addition to the above-identified trends, it is noticeable that early research has focuses on model-based methods, whereas more recent papers explore the possibilities that deep-learning–based approaches offer.

10.4 Front-end

Event-based SLAM systems often consist of several modules, which tackle smaller subproblems, such as feature extraction, data association, bootstrapping, pose estimation, depth estimation, etc. A primary division consists of the front-end and the back-end. From an input-output point of view, the front-end receives the raw sensor data (plus possibly auxiliary information, such as camera calibration) and outputs a set of event camera poses and scene map(s) (see Fig. 10.4). The back-end refines these variables (i.e., the SLAM problem unknowns) to improve the fit between them and the sensor data. It operates after the front-end, at a slower pace (depending on the number of variables involved) and can feed back its output to the front-end to help reduce drift and correct errors.

Therefore, the front-end converts the information from the sensor (e.g., photons) into geometric primitives (e.g., camera poses) and also photometric information (e.g., map appearance). This often comprises a step of "feature" or "information" extraction. Hence, the first challenge is to understand the information contained in the event stream and be able to extract it using methods that preserve the characteristics of the data (low latency, sparsity, HDR, etc.). Assuming constant illumination, events are caused by moving contours (edges). Therefore, we may consider a moving event camera as an asynchronous edge detector, which means that the SLAM problem is formulated in terms of scene contours (Fig. 10.3). This is a priori sensible because contours are the most informative regions of the image plane, allowing us to estimate retinal motion, from which 3D information is inferred. Each event consists only of a 4-tuple and is subject to noise, hence it carries little information; thus many events (e.g., thousands, millions) are needed to produce reliable estimates of quantities such as camera poses and scene maps. Extraction of information from the event stream depends on the task and on many design choices, such as the type of representation of the SLAM variables (scene map, camera trajectory), the hardware used to process the data, the output rate, etc.

10.4.1 Pre-processing. Event Representations

In the SLAM problem, the event camera continuously outputs data as it moves through the scene. Events are triggered "everywhere" on the image plane, as from the camera's point of view it appears that all scene edges are moving. Since events are sparse and have microsecond resolution, each of them corresponds to a different camera pose. This is radically different from traditional (frame-based) cameras, where all pixel measurements of an image have the same timestamp and therefore share a common camera pose (this is the paradigm on which traditional multi-view geometry [444] has been built). Many SLAM methods convert event data into alterantive representations (event images, time maps or "time surfaces", voxel grids, etc.) [369] for different reasons, such as compatibility with conventional computer

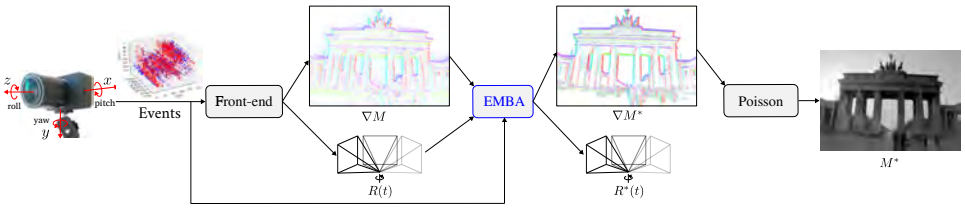


Figure 10.4 Event-based SLAM pipeline with a front-end (that computes and a map and camera poses) and a back-end (that refines the map and poses). Since events respond to moving edges, the recovered map is often an edge (gradient) map. The example shows a direct, rotational SLAM pipeline (poses consists of rotations, and the map reduces to a panoramic map) [424]. An absolute intensity map may be recovered by Poisson integration.

vision methods, easier interpretation, etc. This conversion step often implies a quantization of the information (e.g., grouping events with similar timestamps) and/or a loss of the sparsity (e.g., zero-filling arrays at locations where no events happen).

Therefore, the study of event representations [369, 378] has gained attention. It is typically the first stage of the front-end and it highly influences later processing stages: events are converted into a more familiar representation (e.g., images) that are easier to work with (to feed to mature SLAM methods designed for traditional images, or to design learning-based methods based on images). This conversion is in part due to the fact that the research community is still exploring the best way to extract information from the event stream and tries to reutilize mature image-based methods. The front-end may use different event representations; for example, EVO [921] uses raw events for its mapping module (EMVS [923]) and event (edge-like) images for its camera tracking module. Ideally, one would design SLAM methods that use event representations that preserve the high speed and sparse properties of event cameras and do not suffer from the issues of traditional cameras (quantized time, latency, non-sparsity). In practice, this is an emerging research topic that requires rethinking visual processing asynchronously, and there is still ample room for improvement and investigation of fundamental results.

10.4.2 Indirect Methods

The design choices of the front-end largely influence the rest of the system. A major design choice is the type of processing method: indirect or direct. Indirect methods have broadly two steps; they first extract and track point-based, line-based or other type of feature from the events, and then leverage results from classical SLAM to estimate the camera motion and the structure of the 3D scene based on such geometric primitives. Features compress the event data into few informative primitives, which enables focusing the computational resources. A central problem consists in establishing and maintaining correspondences among the event features

(and the map landmakrs), which is known as data association. This is challenging, as each event carries little information and is motion-dependent to unambiguously determine association. Due to the high temporal resolution of event cameras, association can be established by spatio-temporal vicinity in pixel space. Hence, it is natural to track features rather than to match them.

Camera pose estimation or camera tracking is often formulated as the solution of a feature registration / alignment problem by minimization of a geometric objective (e.g., the reprojection error, measured using Euclidean distance in pixel space) given a map of the scene. The 3D structure of the scene is typically computed by means of triangulation (i.e., back-projection) of corresponding feature locations (e.g., to obtain 3D points and lines) using given camera poses. A large toolbox of mature geometric methods (multi-view geometry [444]) can be exploited.

Like conventional visual SLAM, indirect event-based methods rely heavily on robust feature extraction and tracking. However, these components are not yet as mature as their frame-based counterparts because they have to deal with unique challenges (large noise, sparsity, asynchrony, motion dependence, etc.). This limits the accuracy and therefore applicability of these systems. To address these issues, some systems resort to sensor fusion (with grayscale images and/or IMU data).

10.4.3 Direct Methods

Direct methods use all data available (not just the event data that conform to the definition of a feature) to estimate camera motion and 3D scene structure. They directly align event data with maps, images or other events without explicit feature extraction. If the event rate is high compared to the processing capacity of the system, data reduction mechanisms (e.g., denoising, subsampling, etc.) are adopted to reduce the number of events to process [423, 580].

As direct methods have only one step, the motion (camera tracking) or scene parameters (mapping) are obtained by optimization of some objective function (e.g., photometric error, spatial event rate error, etc.). The photometric-based objective induces a geometric registration objective. The problem unknowns are obtained by the alignment of edge-like brightness patterns conveyed by events and/or corresponding image or map pixels. Direct methods rely on the quasi-continuous nature of event data, for example to compute an incremental camera pose from the previously estimated one: the increment is small, as events are continuously triggered without gaps or blind times.

Among direct methods, a prominent subclass due to their state-of-the-art accuracy performance is that of methods that estimate motion or scene parameters by event alignment, which appears in the form of sharp images of warped events (IWEs). The idea is to estimate motion by "undoing it", i.e., finding the parameters that motion-compensate the event data. Event alignment can be measured by means of different objectives: variance, gradient magnitude, disperson, etc. They

are equivalently known as Focus or Contrast Maximization (CMax) [368, 1016]. In problems where events can be warped to a few pixels or a line, these methods can suffer from undesired global optima [1015]. Data association in direct methods is typically handled implicitly and in a soft manner, inherited by the distance in the pixel grid. Nevertheless, hard associations using nearest-neighbor values are also possible and effective in some cases.

10.4.4 Model-based and Learning-based Methods

So far, the majority of event-based SLAM approaches are hand-crafted, designed by human intuition on principles of operation of the event camera and the SLAM problem. Instead, deep-learning methods leverage artificial neural networks (ANNs) to model event data, either by converting events into image-like representations or by processing them directly with Spiking Neural Networks (SNNs). These methods are often categorized into supervised or self-supervised, depending on the type of supervisory signal. Self-supervised methods rely on events or other sensors (e.g., colocated grayscale images) to estimate depth and camera pose by leveraging some temporal dynamics or photometric consistency loss [1238]; whereas supervised methods require ground truth data for training [593], which is typically difficult to acquire in the real world. In recent years, many multi-sensor datasets have been recorded onboard cars, drones, etc., which can provide the data needed for ANN training.

Learning-based solutions may substitute parts of the SLAM pipeline, such as feature extraction and tracking [767], or try to replace the entire system (end-to-end). Learning-based approaches offer the advantage of handling complex data representations and noise implicitly, but require large datasets for training and may suffer from generalization issues when applied to significantly different scenes (i.e., "domain shift") from the ones in the training set.

10.5 Back-end

The goal of a refinement module like the SLAM back-end [142] is to improve the consistency between the variables of the SLAM problem and the sensor data, thus improving accuracy and robustness of the fit, reducing the propagation of errors between tracking and mapping modules of the system. Often, bundle adjustment (BA) [1111] is used as synonym for back-end.

Event-based BA is still in its infancy, as most event-based SLAM systems lack a refinement step. Instead, they operate in a parallel tracking-and-mapping manner [579, 921, 1297], with each module relying on the output of the other concurrently running module as input to work properly. They have prioritized simplicity and taking advantage of the low-latency benefits of event cameras over accuracy and robustness. In addition to the challenges mentioned in Sec. 10.2 (noise, motion-dependent

appearance, etc.), an event-based back-end poses the challenge of jointly estimating correlated variables, which implies a high-dimensional search space, making optimization costly (in complexity and latency) and prone to local minima.

Only recently the problem of BA has been tackled in systems that include event cameras. Since the back-end of a SLAM system is highly determined by the output of the front-end (as there needs to be a tight integration between both modules for best performance), we categorize event-based back-ends as indirect (feature-based) or direct (photometric-based).

Indirect back-ends are inherited from classical indirect frame-based methods [1111, 652, 793]. They operate on geometric primitives (corners, lines, etc.) that are detected in the event stream (possibly preceded by an events-to-image conversion [212, 949] to reutilize frame-based detectors). The objective typically consists in the minimization of the reprojection error, measured by the Euclidean distance in the image plane [444]. This approach has the advantage of reutilizing mature, robust techniques in classical SLAM. However, it discards the large amount of information contained in the events (as revealed by image reconstruction methods [925, 1275]) and it is not yet effective: due to noise and the dependency of events on motion, current event corners are not as accurate and stable as frame-based ones, hence their use in SLAM has been scarce [619]. Examples of indirect back-ends include [949, 212, 1154].

Direct back-ends work on sensor data (rather than geometric primitives) and the objective typically consists in the minimization of some form of photometric error. Hence they are more tailored than indirect ones. Approaches like [469], which leverage grayscale information from colocated frames, borrow the back-end from frame-based systems [316, 33]. However, grayscale frames can suffer from motion blur and low dynamic range. Event-only back-ends do not suffer from these limitations; they are recent and so far have been developed for constrained motions (planar or rotational). The objective may consist in the maximization of event alignment (also called motion compensation or CMax) [367, 423] or the minimization of the photometric error (i.e., temporal contrast) conveyed by each event [424, 425]. They are designed based on the event generation model (10.1). As each event carries little information and the number of problem unknows in SLAM is typically large, many events are needed for accurate BA, which poses demands on computational resources, power and latency. There is plenty of room for investigatation of efficient direct, event-only BA in natural scenes and 6-DoF motion scenarios.

10.6 State-of-the-Art Systems

Table 10.1 collects concrete systems in event-based VO/SLAM, describing some of their characteristics (direct, indirect, etc.) according to the categorization introduced in previous sections. While it is not possible to describe all of them in detail in this chapter (and neither is our intention), certain trends are worth mentioning.

System	M/DL	I/D	Event represent.	BA	Motion	Scene	Input	Remarks
Cook [237]	M	D	Event Frame	✗	Rot	Natural	E	Interacting network using optical flow
Weikersdorfer [1173]	M	I	Individual Event	✗	Planar	2D B&W	E	First filter-based Ev-SLAM.
PF-SMT [578]	M	D	Individual Event	✗	Rot	Natural	E	Two interleaved Bayesian filters
Censi [166]	M	D	Event Packet	✗	6DoF	B&W	E+F+D	Filter-based VO based on image gradient
EB-SLAM-3D [1174]	M	D	Individual Event	✗	6DoF	Natural	E+D	Augment events with depth sensor
Yuan [1255]	M	I	Event Frame	✗	6DoF	B&W	E+I+M	Vertical line-based camera tracking
Kueng [619]	M	I	Local Point Set	✗	6DoF	Natural	E+F	Event-based feature tracking VO
ETAM [579]	M	D	Individual Events	✗	6DoF	Natural	E	Three interleaved filters
CMax -ω [364]	M	D	Individual Events	✗	Rot	Natural	E	Contrast Maximization
EVO [921]	M	D	Edge Map	✗	6DoF	Natural	E	Event-event geometric alignment
EVIO [1298]	M	I	Point sets	✗	6DoF	Natural	E+I	Filter-based and MC features
Rebecq [922]	M	I	MC Event Images	✓	6DoF	Natural	E+I	Feature-based, sliding-window back-end
RTPT [931]	M	D	Individual Events	✗	Rot	Natural	E	Panoramic tracker and mapper
Gallego [366]	M	D	Individual Events	✗	6DoF	Natural	E+M	Resilient sensor model
Mueggler [785]	M	D	Individual Events	✓	6DoF	Natural	E+I+M	Continuous-time pose estimator
USLAM [949]	M	I	MC Event Images	✓	6DoF	Natural	E+F+I	Sensor fusion & sliding-window back-end
Chin [212]	M	I	Event Frames	✓	Rot	Stars	E	Tailored to star tracking
ESVO [1297]	M	D	Time surfaces (TS)	✗	6DoF	Natural	2E	Stereo matching on TS patches
Hadviger [432]	M	I	Corners on TS	✗	6DoF	Natural	2E	Cross-corr. feature descriptors
CMax-GAE [580]	M	D	Individual Events	✗	Rot	Natural	E	Contrast maximization
EKLT-VIO [732]	M	I	Individual events	✓	6DoF	Natural	E+F+I	EKLT tracker and VIO back-end
EDS [469]	M	D	Event images	✓	6DoF	Natural	E+F	Frame-based back-end (DSO)
CB-VIO [691]	M	I	Individual events	✓	6DoF	Natural	E+F+I	Feature tracker and VIO back-end
Wang [1154]	M	I	Binary images	✓	6DoF	Natural	2E	Feature matching
El Moudni [311]	M	D	Time Surfaces TS	✗	6DoF	Natural	2E	Use ESVO tracker and EMVS mapper
ESVIO [189]	M	I	Time surfaces (TS)	✓	6DoF	Natural	2E+2F+I	Feature tracking on from TS
ESVIO-direct [692]	M	D	Time surfaces (TS)	✓	6DoF	Natural	2E+I	Extension of ESVO
PL-EVIO [418]	M	I	Time surfaces (TS)	✓	6DoF	Natural	E+F+I	Point & line features, sliding-window BA
CMax-SLAM [423]	M	D	Individual Events	✓	Rot	Natural	E	Contrast Maximization refines motion
EVI-SAM [417]	M	D,I	Individual Events	✓	6DoF	Natural	E+F+I	Dense mapping
Zuo [1312]	M	D	Individual Event	✗	6DoF	Natural	E+D	Augment events with depth sensor
DEVO [593]	DL	I	Event voxel grids	✓	6DoF	Natural	E	Event-version of DPVO [1088]
EMBA [424]	M	D	Individual Events	✓	Rot	Natural	E	Refines motion and gradient map
EPBA [425]	M	D	Individual Events	✓	Rot	Natural	E	Refines motion and intensity map
ES-PTAM [389]	M	D	Events (also as frames)	✗	6DoF	Natural	2E	Use EVO tracker and EMVS mapper
ESVO2 [818]	M	D	Time surfaces (TS)	✓	6DoF	Natural	2E+I	Extension of ESVO
DEIO [416]	DL	I	Event voxel grids	✓	6DoF	Natural	E+I	Extension of DEVO and DPVO

Table 10.1 *Summary of Event-based Visual SLAM methods, sorted chronologically. The columns indicate: the type of method (**M**odel-based or **D**eep-**L**earning–based, **D**irect or **I**ndirect), whether the method has a global refinement module (i.e., back-end / BA), the type of camera motions (Rotational, Planar, 6-DoF) and scenes (high-contrast black-and-white, etc.) it can handle, and the type of input data used (**E**vent camera, **F**rame-based camera, **D**epth sensor, **I**MU and **M**ap), where "2E" means stereo events (two sensors).*

The literature is dominated by model-based systems; data-driven approaches have not taken over yet (although that might happen in the near future, as it occurred with other computer vision tasks). Ever since the beginning, the problem of SLAM with event cameras has been tackled under different assumptions, increasing the complexity in terms of (i) camera motions, (ii) type of scenes, and (iii) additional sensors (or information, such as a map of the scene) to simplify the problem (e.g., a depth sensor attached to an event camera decreases the burden of depth estimation from events alone, and IMUs provide accurate angular velocity information, etc.).

Once an event-based method shows good performance, it is incrementally im-

proved in an almost standard "exploitation" roadmap (similar to frame-based SLAM): for example, monocular methods [921] can be extended into stereo or multi-camera scenarios [389], event-only methods like [1297] (resp. [593]) can be robustified using inertial data fusion [692, 818] (resp. [416]), base system can be extended to handle omnidirectional lenses, etc. Despite this "exploitation" path, event-based SLAM is still an emerging field and, therefore, is in an exploration phase (of different techniques). This becomes evident when analyzing the methods in Tab. 10.1: diverse ideas and principles, leading to different map representations, event representations, loss functions, etc., are leveraged to design the estimation algorithms underpining these systems. There is still plenty of room to investigate new state estimation ideas, especially those that take advantage of the genuine characteristics of the sensor.

10.7 Datasets, Simulators, and Benchmarks

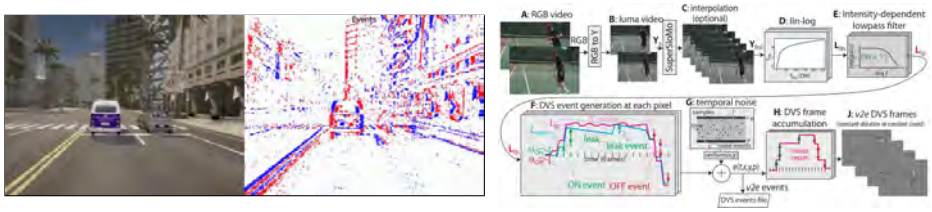
Prototyping, training and benchmarking event-based vision systems places high demands for high-quality, diverse and rich data (real and synthetic). The development of simulators, datasets and leaderboards is essential to move the field forward and establish a common and solid ground in scientific and technical progress. Let us describe prominent SLAM datasets, benchmarks and simulators for event cameras.

10.7.1 Simulators

There are a variety of publicly available tools for generating high-quality synthetic event camera data. ESIM [924] is an evolved version of [784], which was one of the first simulators to mimic the principle of operation of an event camera. Previous efforts, such as [543], just thresholded the difference between two successive frames to create edge-like images that resembled the output of an event camera. ESIM tightly couples the rendering engine and the event simulator, which allows the latter to adaptively render frames based on the dynamics of the visual signal.

The event camera simulator in CARLA [287] expands ESIM in more diverse, rich, and complex scenarios for autonomous driving. In the context of learning monocular depth from events [468], the event camera sensor developed in CARLA takes the rendered images from the simulator and computes per-pixel brightness changes to simulate an event camera in the same way as in ESIM. Figure 10.5 shows RGB images and events generated in the CARLA simulator.

Motivated by learning-based approaches that require large amounts of event data for training and the fact that event data are hardly available due to the novelty of the sensor, a tool for converting any existing video recorded with a conventional camera into synthetic event data was developed: Video to Events (Vid2E) [379]. Hence, Vid2E aims at reducing the gap between publicly available datasets in traditional and event-based computer vision by enabling the use of a virtually unlimited number of existing video datasets for training networks designed for real event data.



(a) CARLA simulator

(b) v2e simulator

Figure 10.5 Event camera simulators: (a) RGB image and generated events using the ESIM simulator in CARLA [287]; (b) Detailed data processing steps of the V2E tool [1277] (TODO: Reuse permission).

Vid2E solves the event data scarcity bottleneck by combining ESIM and adaptive video frame interpolation. ESIM can address this problem by adaptively rendering *virtual* scenes at arbitrary temporal resolution. However, video sequences typically only provide intensity measurements at fixed and low temporal resolution. Super SloMo [528] allows to reconstruct frames at arbitrary temporal resolution and then applies the event camera simulator ESIM. The number of intermediate frames is carefully chosen since, on the one hand, a low value leads to aliasing of the brightness signal, and on the other hand, high values impose a computational burden.

An important aspect of an event camera simulator is to accurately model noise, to reduce the simulation-to-real gap when transferring the algorithms. For example, ESIM, and therefore its derivative simulators, implement a simple noise model based on empirical observation [668]: the contrast threshold of an event camera (C in (10.1)) is not fixed but follows a normal distribution. To simulate this, at each step of the simulation, ESIM samples from a normal distribution $\mathcal{N}(C, \sigma_C^2)$, where the noise level σ_C can be adjusted. Additionally, ESIM allows for separate positive and negative contrast thresholds (C^+ and C^-) to more accurately simulate a real event camera. Other noise effects, such as spatial and temporal variations in contrast thresholds due to electronic noise or the limited bandwidth of event pixels, shall also be considered in event camera simulators.

Vid2E [379] models an ideal event camera lighting. V2E [489] proposes instead a more realistic noise simulator of an event camera based on the DVS circuitry. V2E is the first event camera simulator that includes temporal noise, leak events, and finite intensity-dependent bandwidth, including the same Gaussian threshold distribution as in Vid2E. V2E is a step forward towards a more realistic simulator enabling the generation of synthetic datasets covering a range of illumination conditions, which is an important use case for events. Similarly to Vid2E, V2E uses Super SloMo [528] to increase the temporal resolution of the input video. Figure 10.5 depicts the V2E architecture in detail.

Simulating event camera noises is a challenging topic for realistic synthetic event generation, EventGan [1301] proposes an end-to-end approach using deep learning

to simulate event camera data. Their work proposes a method that leverages the existing labeled data for images by simulating events from a pair of temporal image frames, using a U-Net [940] encoder-decoder network. The methodology consists of training a neural network on pairs of images and events. Instead of applying a direct numerical error loss, they use an adversarial discriminator loss and a pair of cycle consistency losses. EventGAN generates a 3D spatio-temporal voxel grid for each polarity (instead of a set of individual events). This voxel grid representation is commonly used as input to ANNs.

VISTA 2.0 [37] is a simulator that integrates multiple sensor types, including RGB cameras, LiDAR, and event-based cameras, to facilitate policy learning for autonomous vehicles. It uses high-fidelity, real-world data to simulate diverse scenarios, such as varying weather, lighting, and road conditions. The event camera simulator works similarly to ESIM with adaptive sampling. The bidirectional optical flow between two consecutive frames is estimated using an ANN. VISTA 2.0 is designed for training perception-to-control policies, demonstrating enhanced robustness and sim-to-real transfer capabilities compared to real-world training data alone, thereby improving vehicle control in safety-critical situations.

Video to Continuous Events (V2CE) [1277] tackles the problem of producing events with more realistic timestamps than previous simulators. Vid2E and V2E generate events at discrete timestamps, instead of a continuous-time fashion like real events. This is not negligible in tasks that are sensitive to timestamp distribution, which prohibits the use of synthetic events since they can bring a significant domain shift with respect to real events. V2CE simulator works in two stages. The first stage consists of a supervised 3D U-Net encoder-decoder ANN that predicts two voxel grids (one per polarity, similar to EventGAN) from video. The second stage recovers precise event timestamps from the voxel grids. The method iteratively deduces the event count and their relative positions in each voxel. V2CE also shows that it can accurately generate events in saturated light areas and in edges where the event generation model for an ideal sensor does not hold.

10.7.2 Datasets and Benchmarks

The number of event-based datasets dedicated to Visual Odometry and SLAM has increased significantly since the publication of the ECDS [784] (see Tab. 10.2). ECDS was the first dataset with synchronized events, IMU, and ground-truth camera poses in 6-DoF. Previous datasets [960] included both synthetic and real events featuring pure rotational motion (3 DoF) in simple scenes with high visual contrast; ground-truth data was obtained using an IMU. Other work [62] enabled a 5 DoF comparison of event-based and frame-based camera movements; and ground truth was obtained from the pan-tilt unit encoders and the ground robot's wheel odometry, making it prone to drift. ECDS contains hand-held, 6-DoF motion (slow- and high-speed) on a variety of scenes with precise ground-truth camera poses from a

motion-capture system. The dataset consists of 11 scenes with real events and two additional scenes with synthetic events. The synthetic data was produced with the first version of what became the ESIM [924] simulator.

The RPG stereo dataset [1296] consists of eight hand-held sequences recorded with a stereo DAVIS [117] in an office environment and a synthetic sequence (featuring three fronto-parallel planes at various depths) produced by the simulator [784]. Although this dataset does not provide ground-truth depth, it has accurate ground-truth poses from a motion capture system and serves as a good starting point for prototyping and evaluating event-based stereo SLAM methods.

The Multi Vehicle Stereo Event Camera Dataset (MVSEC) [1299] is the first dataset to offer ground-truth depth across a variety of platforms. It captures both indoor and outdoor scenes with varying levels of illumination and movement speeds. The platforms include a handheld rig, a hexacopter, a car, and a motorcycle, all equipped with calibrated sensors like 3D Lidar, IMUs, and standard frame-based cameras. MVSEC has wide-ranging applications in pose estimation, mapping, obstacle avoidance, and 3D reconstruction, offering accurate ground-truth depth and pose data through its integrated Lidar system. The datasets contain long sequences that enable a comprehensive evaluation of event-based algorithms.

The UZH-FPV (First Person View) dataset [265] is specifically designed to advance research in autonomous drone racing. It features a custom-built quadrotor with a Qualcomm Flight Board and an mDAVIS346 [1078] event camera mounted on a Lumenier QAV-R carbon fiber frame. The recordings capture indoor and outdoor scenes at varying speeds and trajectories, which present challenges for navigation and state estimation. This dataset supports research in VIO, event data processing, and real-time drone applications in fast scenarios. It has become a key resource for developing high-speed camera motion algorithms, particularly in autonomous drone racing, and has also been used for competitions at conferences and workshops.

The Event Camera Motion Segmentation Dataset (EV-IMO) [775] is the first event-based dataset created specifically for segmentation of independently moving objects (IMO) in indoor environments. It contains 32 minutes of recordings, tracking up to three fast-moving IMOs using a motion capture system. The dataset provides pixel-wise motion masks, and ground-truth egomotion and depth. It is useful in robotics, especially in scene-constrained environments where accurate motion detection is crucial for tasks like object tracking and autonomous navigation. EV-IMO2 [132] builds on its predecessor by offering more sequences, three higher quality event cameras, and more complex scenarios. This version serves as both a challenging benchmark for current algorithms and a rich training set for developing new methods, including event-based SLAM in monocular and stereo setups.

The Stereo Event Camera Dataset for Driving Scenarios (DSEC) [380] is large-scale, intended to support research in autonomous driving, especially in developing robust perception systems capable of handling adverse lighting conditions through sensor fusion of events and frames. The dataset features a platform with a multi-

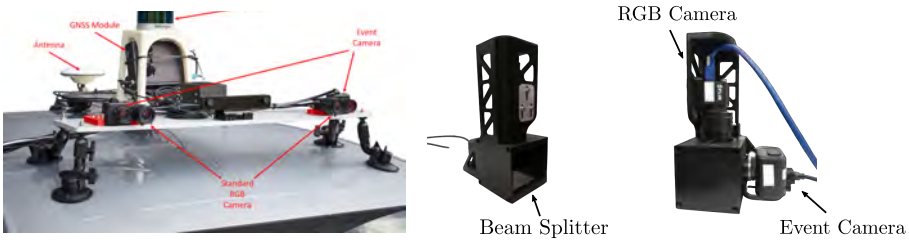
Dataset	Platforms	Pixel Resolution	Sensors
ECDS [784]	Hand-held	240×180	E, F, I
RPG-stereo [1296]	Hand-held	240×180	2E
MVSEC [1299]	Hand-held, Drone, Car, Byke	346×240	2E, 2F, I, Lidar, GPS
UZH-FPV [265]	Drone	346×260	E, F, I
EV-IMO [775]	Hand-held	346×260	E, F, I, Depth
EV-IMO2 [132]	Hand-held	640×480	3E, F, I, Depth
DSEC [380]	Car	640×480	2E, 2F, Lidar, GPS
TUM-VIE [592]	Hand-held	1280×720	2E, 2F, I
EDS [469]	Hand-held	640×480	E, F(RGB), I
VECtor [371]	Hand-held	640×480	2E, 2F, RGB-D, I, Lidar
M2DGR [1245]	Ground Robot	640×480	E, F, I, Lidar, GPS, Thermal
ViViD++ [642]	Hand-held, Car	240×180, 640×480	E, F, RGB-D, Thermal, Lidar, GPS
FusionPortable [530]	Hand-held, Quadruped Robot	346×240	2E, 2F, I, Lidar, GPS
Stereo HKU-VIO [189]	Hand-held	346×260	2E, 2F, I
M3DE [169]	Drone, Car, Quadruped Robot	1280×720	2E, 2F, I, Lidar, GPS
CoSEC [868]	Car	1280×720	2E, 2F, I, Lidar, GPS

Table 10.2 *Summary of event-based SLAM datasets, sorted chronologically. Same notation for sensor data as in Tab. 10.1. Stereo and multi-sensor datasets are further described in the survey [388].*

camera setup, including two VGA-resolution event cameras (Prophesee Gen 3.1) with a 60 cm baseline and two RGB cameras (FLIR Blackfly S) with a 51 cm baseline (see Fig. 10.6). The setup includes a Velodyne VLP-16 lidar and an RTK GPS for precise localization. Data were collected in various urban and rural settings in Switzerland under diverse illumination conditions, such as day, night, and direct sunlight, providing ground-truth depth maps for stereo matching. DSEC also provides Optical Flow and Disparity to benchmark algorithms in challenging driving conditions. These benchmarks use metrics like N-pixel disparity error, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE) to assess the performance of algorithms combining high-resolution event camera data with RGB frames.

The TUM Stereo Visual-Inertial Event Dataset (TUM-VIE) [592] employs stereo Prophesee Gen4 event cameras (1 Megapixel resolution) along with synchronized IMU data at 200Hz and stereo grayscale frames at 20Hz. It includes sequences from handheld and head-mounted setups in diverse indoor and outdoor environments, covering various scenes such as sports activities, HDR scenarios, and low-light conditions. TUM-VIE is intended to facilitate research on VIO, SLAM, 3D reconstruction, and sensor fusion, especially in challenging conditions where traditional methods may fail, pushing the boundaries of high-resolution event-based perception algorithms.

The Event-Aided Direct Sparse Odometry (EDS) dataset [469] includes high-quality events, color frames, and IMU data to support research in monocular VIO. Data were acquired using a custom-made beamsplitter device (see Fig. 10.6), allowing for precise alignment of RGB frames and events on the same optical axis, which is not commonly found in previous datasets. The scenes recorded include natural in-



(a) DSEC (b) EDS

Figure 10.6 Details of some Event-SLAM datasets: (a) the sensor suite mounted on top of a car, in the DSEC [380] dataset (©2021 IEEE). (b) Details of the beamsplitter that allows the sensors to share a spatially aligned field of view in the EDS [469] dataset (©2022 IEEE).

door environments, providing high-resolution, well-calibrated data for applications like optical flow estimation, depth estimation, and robust visual odometry under various motion and lighting conditions.

The Versatile Event-Centric (VECtor) Benchmark Dataset [371] is also designed to evaluate event-based SLAM algorithms. The recording platform holds a diverse sensor suite, including stereo cameras (event- and frame-based), an RGB-D sensor, a 128-channel LiDAR, and a nine-axis IMU, all mounted on a versatile 3D-printed holder. The dataset features both small-scale indoor environments, like a motion capture arena, and large-scale indoor environments with various complexities and illumination conditions. It claims to offer high-resolution (VGA), synchronized data across diverse environments, ensuring reliable evaluation of SLAM algorithms in both static and dynamic, low-light, and HDR scenarios. This makes it a comprehensive resource for advancing research in multi-sensor SLAM applications.

The Vision for Visibility Dataset (ViViD++) [642] was recorded with a multi-sensor platform, including thermal cameras, to support research on SLAM algorithms that can handle poor visibility, motion disturbances, and appearance changes, leveraging the complementary strengths of different sensors. The Fusion-Portable dataset [530] includes a Quadruped robot that moves in various scenes, such as corridors, canteens, roads, and gardens under different lighting conditions.

Finally, the Multi-robot, Multi-Sensor, Multi-Environment Event Dataset (M3ED) [169] (informally known as MVSEC 2.0) is focused on high-speed dynamic motions in robotics applications. It combines 1 Megapixel stereo event cameras, grayscale and RGB cameras, a 64-beam LiDAR, and high-quality IMU, all synchronized with RTK localization. Unlike previous datasets, M3ED offers heterogeneous data from multiple platforms in both structured and unstructured environments, with ground truth pose, depth, and semantic labels, making it a valuable resource for developing robust event-based perception algorithms for dynamic environments beyond traditional driving or indoor applications.

10.7.3 Metrics

Ideally, SLAM systems should characterize the quality of their localization and mapping modules individually. However, because (i) both modules operate in an intertwined way (depth errors affect camera pose errors, and vice versa), and (ii) ground-truth localization information is considerably more compact (6-DoF) and easier to acquire than accurate ground-truth depth, the result is that depth estimation errors are *subsumed* in the evaluation of camera trajectory errors.

Conceptually, since both classical SLAM and event-based SLAM output camera trajectories, event-based SLAM inherits the performance evaluation protocol from classical SLAM. Two commonly used metrics are the Absolute Trajectory Error (ATE) and the Relative Pose Error (RPE) [1050]. The ATE assesses the accuracy of the camera's pose relative to a fixed world coordinate system; hence, it provides a broad assessment of the VO system's long-term performance. The RPE evaluates the relative poses between consecutive (i.e., nearby) timesteps; hence, it focuses on the local consistency of VO system. The translational error in ATE, also known as positional error, is calculated as the Euclidean distance between the estimated and ground-truth camera positions. The rotational error, or orientation error, is determined by the geodesic distance in $SO(3)$. Similarly, the translational and rotational parts of RPE are calculated between pairs of camera poses over a time interval. Some studies also compute the positional error relative to the mean scene depth or total distance traveled, ensuring that the error remains invariant to the scale of the scene or trajectory.

Additional error metrics –Average RPE (ARPE), Average Relative Rotation Error (ARRE) and Average Endpoint Error (AEE)– may be used to assess the estimated translation vectors and rotation matrices [1300]. Specifically, ARPE and AEE measure differences in position and orientation between two translation vectors, while ARRE calculates the geodesic distance between two rotation matrices.

Beyond these metrics, average linear and angular velocity errors can also be useful for evaluating camera pose estimation, especially when working with event cameras, where abrupt and fast camera motions are estimated thanks to the events. Camera poses are functions of both velocities over time. Several toolboxes [413, 1278] are publicly available to facilitate the reproducibility of research and reduce the complication in SLAM trajectory evaluation.

In case depth estimation is evaluated separately, the average depth error at various cutoffs up to fixed depth values is often used, allowing for comparison across methods on different scales of 3D maps. However, there are not many datasets that contain ground-truth depth information (see Sec. 10.7). The Root Mean Square Error (RMSE) of the Euclidean distance between the estimated 3D point with respect to the closest surface on the ground truth map is the preferred metric. Additional metrics, such as the Relative Error (REL) and completion (number of points recovered), are also used in the literature [310, 468, 389].

10.8 Outlook

Although research on event-based SLAM has made considerable progress, many open questions and problems remain given the novelty of the technology. These questions pertain to what are the best ways (hardware and software) to acquire and process visual information for a given task (e.g., SLAM) in order to rival or surpass the performance (in terms of robustness, latency, efficiency, accuracy, etc.) observed in biological species.

The sensor is asynchronous, but most of the systems in the literature are designed on serial (i.e., von Neumann) processors (due to the entry barrier to neuromorphic computing). This is suboptimal in terms of efficiency (power consumption), latency, etc. compared to the expected performance of fully neuromorphic systems [851], where event cameras are paired with asynchronous (brain-inspired, spike-based) processors, controllers, actuators, etc. It is a long-standing dream of the research community: to build robots that mimic the efficient processing of animals and their ability to map and localize themselves in the environment (with potential applications in "always on" inside-out tracking for AR/VR, etc.). This dream requires rethinking and co-designing sensors, processors, and algorithms [254] in a neuromorphic engineering way, which is very challenging, as it takes great breadth and depth of expertise, and coordination of multiple disciplines.

In the near future, novel hybrid sensors are being developed that provide data inspired by the two visual streams [1233], spatially and temporally aligned, with low latency, HDR, and fine details (pixel count). Alternatively, foveated sensors [325], mimicking biology, are also investigated to reduce bandwidth requirements. There is still a big field to explore in terms of event cameras, their evolution (e.g., near-sensor processors like pixel processor arrays, Aeveon sensors), and their combination with other sensors (frame-based cameras, structured light, LiDAR, RADAR, etc.) for data fusion and improved SLAM performance.

Acknowledgment

The authors thank Giovanni Cioffi for his support in preparing this chapter.