

Neuromorphic Perception and Navigation for Mobile Robots: A Review

A. Novo^{a*}, F. Lobon^b, H.G. De Marina^a, S. Romero^a, F. Barranco^a

^aDepartment of Computer Engineering, Automation and Robotics, Research Centre for Information and Communication Technologies (CITIC-UGR), University of Granada, 18014, Granada, Spain

^bInstituto de Astrofísica de Andalucía (IAA-CSIC), 18008, Granada, Spain

Abstract

With the fast and unstoppable evolution of robotics and artificial intelligence, effective autonomous navigation in real-world scenarios has become one of the most pressing challenges in the literature. However, demanding requirements, such as real-time operation, energy and computational efficiency, robustness, and reliability, make most current solutions unsuitable for real-world challenges. Thus, researchers are forced to seek innovative approaches, such as bio-inspired solutions. Indeed, animals have the intrinsic ability to efficiently perceive, understand, and navigate their unstructured surroundings. To do so, they exploit self-motion cues, proprioception, and visual flow in a cognitive process to map their environment and locate themselves within it. Computational neuroscientists aim to answer “how” and “why” such cognitive processes occur in the brain, to design novel neuromorphic sensors and methods that imitate biological processing. This survey aims to comprehensively review the application of brain-inspired strategies to autonomous navigation, considering: neuromorphic perception and asynchronous event processing, energy-efficient and adaptive learning, or the imitation of the working principles of brain areas that play a crucial role in navigation such as the hippocampus or the entorhinal cortex.

Keywords: navigation, hippocampus, neuromorphic sensors, brain-inspired

1 Introduction

Robotics, more than an engineering, is an Art. From the earliest robotics breakthroughs, nature has been a fundamental inspiration to pursue the paradigm of the reliable autonomous system. Bio-mimetic mechanical designs [175], neuromorphic sensors [112], swarm behavior policies [50], or even algorithms that imitate the behavior of brain regions [197] are the focus of intensive research in robotics. Autonomous robots perform their tasks within a certain level of self-sufficiency, avoiding, to some extent, human intervention [143]. Most early autonomous robots required an on-purpose configuration of their environment, with e.g. markers or beacons [152]. Such a setup constrains the robots to a specific environment, with a costly preparation. Today, autonomous robots are asked to interact with dynamic and unstructured environments [176], typically without following a fixed script. Indeed, current autonomous robots perceive their environment to sense, plan, communicate, and finally actuate on their environment, usually requiring changes on it [41].

The novel field of neuromorphic engineering investigates and exploits the potential of sensors, computing platforms, and algorithms attempting to mimic biological capabilities. This area emerges in an attempt to achieve what nature proves: how animals effectively perceive, plan, and act, in an energy-efficient and robust fashion and adapt to changing circumstances in real-time. All these properties are of great value for robotics. Attention has also been drawn toward how the brain internally works to accomplish navigation tasks, the brain regions involved in such processing, and their working principles. Although multiple areas are needed, the process mainly occurs in the hippocampus and the entorhinal cortex areas. These areas make possible positioning in a map of the environment with minimal effort [4]. Also, they enable path integration for autonomous navigation. Taking motion cues such as the direction and speed of the individual over time, path integration is able to estimate its position within the map of the scenario. Compared to traditional methods where this process is computationally expensive, our brain does it with minimal resources in such a way that we are not even conscious of that. This is thanks

*Corresponding author
Email address: martineznovo@ugr.es

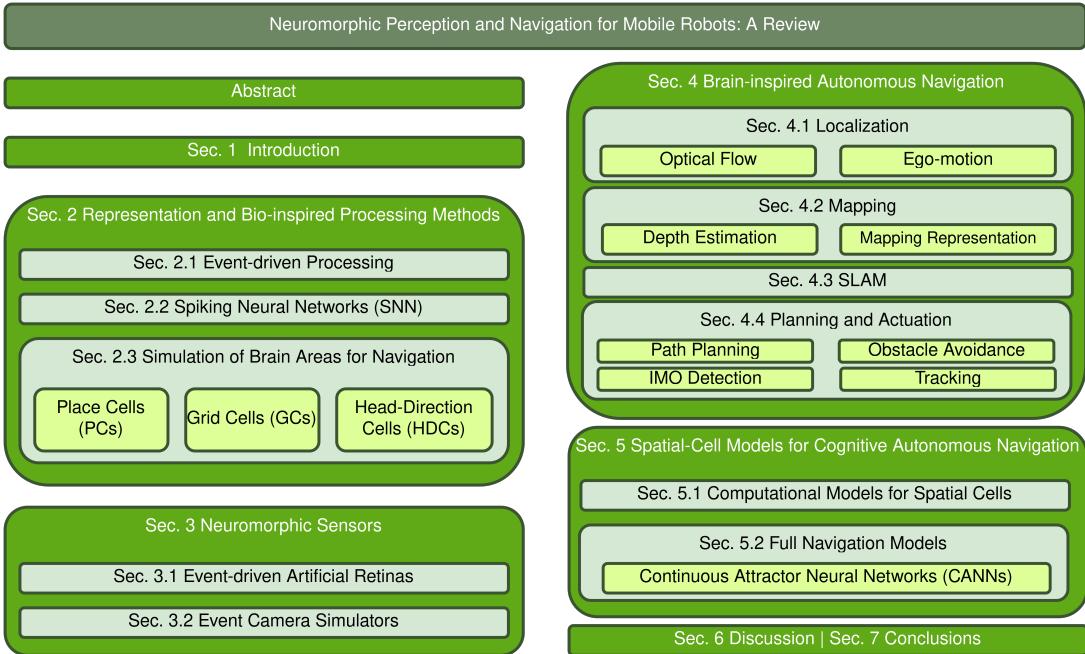


Figure 1: Structural and hierarchical taxonomy of the survey.

to the cognitive map that our brain internally builds when certain groups of cells produce a firing pattern that encodes our location within that map [19]. From this cognitive map, experiments with rats proved that they are able to take shortcuts when navigating, showing that this mental map representation does not only take geometrical information of the outer world, but they also appeared to record information about the events that the animals experienced at specific locations. This capability may be useful in building a new paradigm for the well-known traditional planning methods, where the input mapping representation is crucial for deriving real-time optimal paths based on events for autonomous robots.

Motivation: Neuromorphic devices that mimic biological neural systems at different levels like sensing, communication protocols, architectures, or strategies have been the focus of intense research for decades [179]. On the one hand, engineers have taken inspiration from how the brain works and solves very complex problems in the real world to develop new algorithms and devices. On the other hand, neuroscientists have tried to understand the functionality of the different neural circuits in the brain. Despite the great advances in the last decades, the promise of a massively parallel model based on simple computational units for low-power and real-time computing that mimics biological neural systems with state-of-the-art accuracy remains unfulfilled [43][163]. The review scope is limited to vision-based asynchronous event processing since they are the most popular solutions thanks to the wide availability of neuromorphic vision sensors.

Precisely, the main purpose of this survey is to review how the scientific community faces the problem of autonomous navigation and to characterize the existing solutions according to brain-inspired principles. Moreover, we are committed to shed light on their potential applications, discussing the advantages and limitations of these solutions when used in the real world, compared to conventional approaches. Each robotic navigation subsystem (i.e. localization, mapping, or planning) is individually discussed, with a focus on works that are purely bio-inspired. The objective of this article is to respond to the question *“Could neuromorphic engineering solutions succeed the current and classical navigation approaches in robotics?”*

Outline: Next, we describe the representation and processing of sensory information from a biological perspective in Section 2. In Section 3, sensors based on neuromorphic vision, commonly known as *event cameras*, are described. The core of this review is addressed in Section 4, which covers bio-inspired navigation methods, presenting the whole autonomous navigation pipeline in separate subsystems that are distributed in subsections. A final conceptualized flow diagram of this autonomous pipeline can be found in Fig. 4. Then, Section 5 discusses some specific biological approaches based on brain areas known to be involved in navigation. Finally, we present the discussion and conclusions in Section 6 and Section 7.

2 Representation and Bio-Inspired Processing Methods

Traditional approaches based on mathematical models designed from specific domain knowledge are very efficient due to their simplicity; typically, they avoid the need for large amounts of data as required in Machine Learning approaches. However, these traditional models use statistical or geometrical methods based on measurements of a deterministic domain representation and thus, do fail with high-dimensional dynamic problems [166]. In this survey, these domain-knowledge-based methods are simply referred to as *model-based approaches* as opposed to *learning-based approaches*. However, this survey is focused on brain-inspired methods and thus, we pay special attention to strategies that use neuromorphic data: event-driven processing, adaptive and robust learning with Spiking Neural Networks, and models that imitate the way in which some of the main brain areas involved in navigation work.

2.1 Event-driven processing

State-of-the-art visual processing is based on intensive computing fed with sequences of synchronous images. Differently, biological solutions use continuous streams of asynchronous events that are processed by hierarchical structures to extract features from the scene. This asynchronous independent processing is called event-driven and enables real-time and low-power computation [28].

Events are triggered when a certain threshold is crossed and are emitted as a tuple with 2 fields, using the Address-Event Representation (AER): a) the event *address* corresponds to the coordinates within the sensor matrix that triggered it and b) the *time* of event creation. AER is robust to aliasing, potentially parallelizable, and composable with the ability to easily build hierarchical structures [173]. Unlike frame-based representations, the acquisition is triggered by spatio-temporal changes, and thus, the captured information better represents scene dynamics.

Most approaches to processing neuromorphic data often use well-known solutions that adapt conventional methods. From an engineering point of view, we believe these solutions are valuable if they represent an advancement in the field without ignoring the benefits of event data. However, approaches that merely transform events into artificial frames to be processed by conventional methods, ignoring the scene dynamics, do not take full advantage of the high-temporal resolutions of event cameras. Moreover, building artificial frames from events yields processing redundant information, increasing the required computing resources and data bandwidth.

2.2 Spiking Neural Networks

SNNs are the learning counterpart in neuromorphic computing to conventional neural networks, also known as Artificial Neural Networks (ANNs). ANNs, and specifically the successful DNNs (Deep Neural Networks), have become very popular to solve problems in all different fields. However, the metric of success has exclusively been focused on accuracy, neglecting other qualities such as computational complexity, energy consumption, or storage needs [44]. SNNs emerged to model the behavior of biological neuron networks, where the neuron usually integrates its input spike voltages (action potentials), firing a new spike when the accumulated voltage crosses a threshold (membrane potential). Information is rate- or time-coded via spikes, making transfer fault-tolerant, meaning that missing an individual spike does not significantly impact the average firing rate. Also, given the sparsity of the inputs, SNNs only process information when there are spikes, resulting in reduced computational load and energy consumption compared to traditional continuous computation in ANNs.

SNNs have proven their potential to achieve precision results on par with ANNs while reducing computational cost and energy consumption [44]. However, their main drawback is the lack of a learning rule similar to backpropagation in the powerful supervised learning in ANNs, which enables efficient training. To apply backpropagation for gradient-descent methods, continuous differentiability is required (spikes are not). Some approaches to overcome this limitation are: 1) to transfer weight values from ANNs to SNNs with similar architectures [48]; 2) to adapt backpropagation to SNNs, e.g. using Surrogate Gradient Learning [135] achieving great accuracy gains; 3) or to train using bio-inspired rules such as Spike-Timing-Dependent Plasticity (STDP), a rule that excites the synapse connecting two neurons if their triggers are correlated and inhibits them otherwise, enabling real-time adaptive learning [47].

Finally, encouraging hardware development in the last years has made SNNs the focus of interest for the industry, with platforms such as Intel Loihi [45], IBM TrueNorth [2], or Brainchip Akida [180].

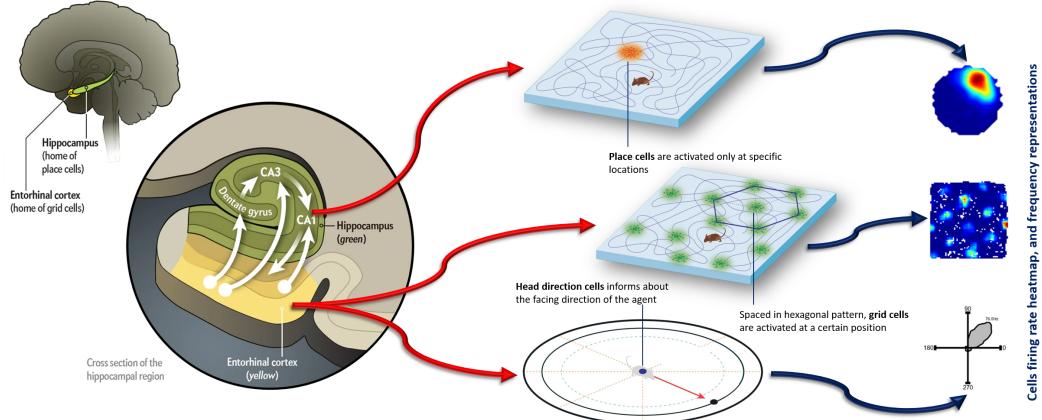


Figure 2: The underlying brain navigation cells process (adapted from [76], [4], and [40]).

2.3 Simulation of brain areas for navigation

Several areas in the brain are known to be involved in cognitive processes regarding navigation, being the most relevant ones the hippocampus and the entorhinal cortex. These areas contribute to building cognitive map representations of the spatial environment or the location of the individual in that map [76, 187]. Neuroscientists point out that navigation is possible thanks to electrophysiological correlates of various types of cells in the brain concretely, three main types that contribute to the brain spatial representation and navigation support [133]:

- Place Cells (PC): They contribute to building a cognitive representation of the space. These cells maximize their firing rate when the animal is in one specific region, namely *Place Field*, regardless of its orientation and what it sees [90]. PCs are thought to be involved in recognizing a previously visited scenario and in route planning [121].
- Grid Cells (GC): These cells are thought to cooperate for self-motion calculation, which is indeed part of the path integration process. Also, they are thought to be part of the pose estimation process while the agent navigates [131].
- Head Direction Cells (HDC): These cells fire for a specific orientation, maximizing their firing rate at a *preferred direction* independently of the agent position. These cells are believed to fully encode the pose of the agent in the environment [172].

Regarding the environment representation, space can be represented as 1) an *egocentric* map where space is defined concerning the agent itself, 2) an *allocentric* representation defined by the relationship between external space features captured during navigation, or 3) an *inertial* map where location and orientation are defined relative to the direction and distance traveled from an initial reference point. The discovery of HDCs led to the theory that these cells are doing integration of head angular velocity, and thus that these brain areas perform *path integration* to build their internal coordinate system with the additional involvement of GCs. In other words, *path integration* is performed using egomotion cues such as optical flow and vestibular signals, to reconstruct the relative pose and distance traveled from the starting point [132]. Moreover, *place recognition* is another critical cognitive process with experiments that suggest that visual information is merged at the level of the entorhinal cortex (EC) [67] as a cognitive map. PCs are believed to do so by merging the “what” and “where” information related to visual stimuli, even when the environment changes dramatically. Eventually, this last is a key feature for Simultaneous Localization and Mapping (SLAM) algorithms in autonomous robotics. Refer to Section 5 for a detailed review of the computational models that replicate the working principles of these brain areas.

3 Neuromorphic Sensors

Neuromorphic sensors are a type of electronic circuit design that mimics biology, in an attempt to take advantage of the efficient cognition processes evolved to improve their performance for millions of years. Applying this so-called neuromorphic engineering to machines opens up new potential to improve their efficiency, latency, or power consumption.

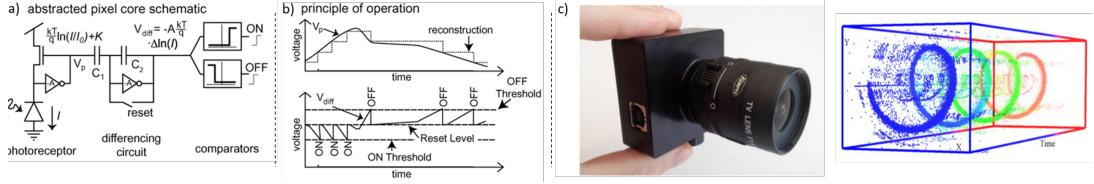


Figure 3: a) Event camera pixel schematic, and b) principle of operation (from [112]). c) Time-space accumulation of events triggered by an event camera for a spinning object.

During the last decades, many neuromorphic sensors have been developed: silicon cochleas that convert vibrations in the air into event-driven signals used, e.g., for sound localization [114]; olfactory sensors that identify volatile chemical components in the air [37]; or spiking tactile sensors that consist of mechanoreceptors that perceive physical distortion and thus shapes or texture, with applications in robotic grasping or prosthetics [21]. However, given the dominance of visual perception [94] most works in the literature are devoted to event-driven artificial retinas.

3.1 Event-driven artificial retinas

Event cameras are imaging sensors that do not capture synchronous frames as conventional cameras do. Every single pixel of the event sensor responds independently to the luminance of the scene, triggering events asynchronously when significant changes occur. Moreover, since each pixel is processed independently, the temporal resolution is very high, with latencies in the order of a few microseconds. In some manner, event cameras naturally perform a smart compression of the scene triggering events only for moving edges in the scene, at a very high speed. All these features hint at the potential of event cameras as very promising sensors for navigation tasks [65].

In detail, for each pixel, a reference log intensity level $L \approx \log(I)$ is stored and compared to the current value at every new timestamp. If the difference goes over a predefined threshold $\Delta L = C$, an event is triggered for this pixel location as a tuple (\mathbf{x}, t, p) , where \mathbf{x} corresponds to the 2D pixel location, t is the timestamp at which the event was triggered, and p the polarity or sign of the log intensity change. Eq. 1 shows this working principle, where ∇L is the log intensity gradient.

$$-\langle \nabla L, \mathbf{x} \Delta t \rangle = C \quad (1)$$

Briefly, the output is an asynchronous stream of events triggered by the relative changes in the scene luminance at a very high speed. The main features of event cameras are:

- **High dynamic range (HDR):** Event cameras have an HDR of up to 120 dB (conventional cameras over 60 dB). This is an advantage under challenging lighting conditions such as low illumination or sudden contrast, as biological retinas do. In navigation, this is relevant to reduce artifacts when e.g. driving out of a tunnel or entering a room.
- **Low latency:** Contrarily to conventional cameras which need to wait for a global exposure time to capture a frame, each pixel value of an event camera is independently processed and transmitted with sub-millisecond latencies. Thanks to this property, event cameras are not severely impacted by motion blur.
- **Low-power consumption:** Since event cameras do only capture luminance changes, areas with redundant static information do not trigger any new events. This usually leads to neuromorphic devices consuming less energy (in the order of a few microwatts vs. tens of milliwatts for conventional sensors). This is another strong point for using this technology in autonomous robots with limited battery power.

Finally, event-based sensors are well-suited for applications where dynamics play a significant role, as happened in navigation. As shown in [46], conventional approaches based on full frames mostly extract spatial features and need to reconstruct the dynamics from the sequences of frames whereas event-driven approaches are the best fit for extracting dynamic spatiotemporal features, which are the key features for navigation processing. Despite all these advantages, the sparsity of event inputs makes effective event-based navigation still a challenging open problem.

In addition, some of the most prominent hardware companies that specialize in event sensor manufacturing are **invation** with their popular series named DAVIS sensors [24], **Samsung** with their neuromorphic DVS sensors [170], and **Prophesee** and their ATIS sensors [57].

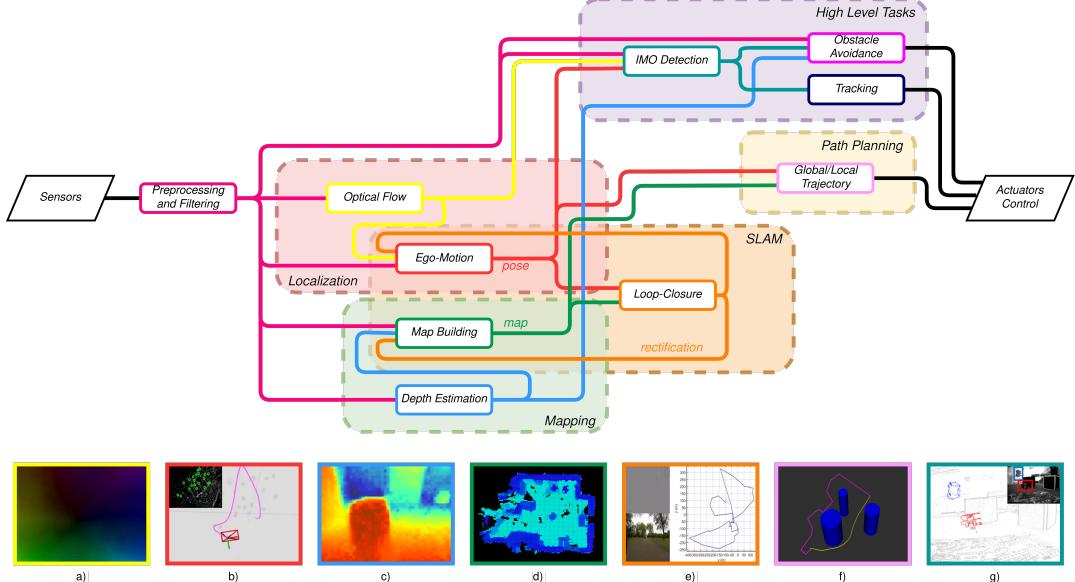


Figure 4: Autonomous navigation flow diagram. Top: For each subsystem, the left arrows are inputs and the right ones are outputs. Bottom: examples of subsystem outputs: a) Optical flow [72], b) Ego-motion [156], c) Depth [146], d) Mapping [87], e) SLAM [125], f) Path planning [138], g) IMO detection [141].

3.2 Event camera simulators

With the success of event cameras and the adoption of deep learning solutions, new large datasets are needed for the advancement of the field. Moreover, event cameras are expensive, and commercial systems are still prototypes, suffering from limitations in their resolution or interfaces. These are the main reasons for building event camera simulators that face challenges in efficiently obtaining realistic data.

ESIM [150] was one of the first complete simulators released as an open platform. This simulator used adaptive rendering to simulate the asynchronous nature of event cameras, providing large amounts of event data. The validation is done by training a network for optical flow estimation with simulated data and evaluating its generalization capabilities when inferring real events. Another very popular simulator is V2E [89], which uses a similar methodology for validation but focuses on the realism of their simulated events. V2E uses a deep learning solution for video interpolation to increase the temporal resolution of the source videos. Both simulators were also compared and extended into the ICNS model [93], which notably improves the noise simulation, and it is also validated with a real sensor. In [168], the ViBES simulator was used to mimic the Object-Motion-Sensitivity (OMS) function of the amacrine and retinal ganglion cell layers [5, 195]. Its goal is to reduce the dimensionality of visual information in each cell to a more encoded feature-rich representation. Finally, [201] presented SpikingSIM, a simulator that generates spiking data from images recorded by conventional frame cameras, including realistic spike noise.

4 Brain-inspired Autonomous Navigation

Guidance, Navigation, and Control (GNC) is an engineering branch focused on the motion of, mostly autonomous, vehicles in their environments. In this survey, we limit our interest to the navigation part. Navigation refers to the measurement and estimation of the robot state vector that, depending on the system, is usually composed of the pose and velocity (and sometimes complemented with acceleration, attitude, rotation rates, etc). Navigation usually includes the sensor selection and modeling, the estimation of the deviations concerning the reference trajectory, and the planning and execution of actions to reduce these deviations between the actual and reference trajectories. Autonomous navigation is usually implemented by combining visual sensors, inertial units, laser, LiDAR, or Global Positioning System (GPS). However, we will focus on visual navigation techniques.

In the next subsections, we cover neuromorphic approaches to build maps and localize agents within them. We also review the most popular path-planning methods, as well as other related tasks such as obstacle avoidance or the detection of independently moving objects (IMOs). Fig. 4 shows a flow diagram for the pipeline of autonomous navigation, highlighting the main components and their relationships, and also illustrating the processing with some results from reviewed methods.

4.1 Localization

Localization enables the continuous tracking of the robot’s position and orientation while it moves through the environment, thanks to sensory inputs. However, these sensory inputs are noisy, which leads to incorrect position and orientation estimates and ambiguities. The estimation can be refined by combining additional cues relative to the scene landmarks, the motion of the objects in the scene (optical flow), and the robot’s self-motion (ego-motion), which are provided by vision sensors.

4.1.1 Optical Flow

Optical flow determines each pixel velocity in the 2D image plane. Thus, it can be directly linked to the robot’s speed and consequently to its state vector. Traditional optical flow estimation algorithms cannot be applied, at least directly, to event data, as they usually rely on the minimization of the frame-to-frame pixel brightness, also known as brightness or Optical Flow Constraint (OFC). Event cameras do not produce brightness information, but only events when changes in the brightness for the same location reach a certain threshold. However, event information is substantially more valuable, since it encodes the spatiotemporal dynamics of the scene, focusing on edges that tend to be more robust for flow estimation. Furthermore, image-based methods struggle in the presence of artifacts such as saturation or motion blur because of the limited dynamic range of conventional cameras, or their limited temporal resolution. High dynamic range and high temporal resolution are exactly two of the advantages that event-driven methods can exploit if using neuromorphic sensors as mentioned in Section 3.1.

Event model-based approaches can be categorized into frame-based Lucas-Kanade [17, 158], filter-bank-based [12, 25], spatio-temporal plane-fitting [1, 18, 158], variational [9], and contrast maximization [165] methods. Apart from them, the most recent approaches are learning-based, using either ANNs or SNNs. The most common architectures are U-Net [155], FireNet [162], and long-term memory-based networks [174]. These learning-based methods are categorized into supervised, self-supervised, or unsupervised approaches.

Next, we describe in detail the most relevant works in the literature for optical flow, distinguishing between model-based methods, deep learning approaches using conventional ANNs, and SNN-based solutions, and summarize the most relevant details in Table 1.

Traditional Methods. The first approximation for optical flow estimation is to build artificial frames to be processed by conventional methods. However, this strategy sacrifices the low latency and high-temporal resolution of asynchronous events with the burden of computing a large number of artificial frames as conventional methods do. In practice, most traditional methods start with the OFC [117]. Assuming no illumination changes (at time t and $t + 1$), the pixel brightness I remains constant. Therefore, the OFC can be written as $I(\mathbf{x}, t) - I(\mathbf{x} + \mathbf{u}, t + 1) = 0$, where the motion of pixel \mathbf{x} is $\mathbf{u} = (u, v)$. Applying the first-order Taylor series expansion, the OFC is expressed as $\nabla I \cdot \mathbf{u} = -I_t$. Extending it, the equation can be rewritten in its most familiar form $I_x u + I_y v + I_t = 0$, with I_x, I_y, I_t the spatial and temporal partial derivatives. Since the equation has two unknowns u, v and only one equation, it is usually extended with new constraints to solve it. For example, Horn and Schunck method [86] assumes that the optical flow is spatially smooth, solving for \mathbf{u} by minimizing the expression in Eq. 2.

$$\arg \min_{\mathbf{u}} \int_{\Omega} \lambda^2 (\|\nabla u\|^2 + \|\nabla v\|^2) + (\nabla I \cdot \mathbf{u} + I_t)^2 d\mathbf{x} \quad (2)$$

In this expression, the error includes the regularization term which serves to smooth the flow, and the data term in this case is the original OFC, with λ^2 as the smoothness weighing factor.

Lucas-Kanade. In [17] authors proposed the first approach that naively adapts the Lucas-Kanade method to events, approximating spatial and temporal partial derivatives as the derivatives of the accumulation of events at a pixel for a period of time. This approximation generally fails when edges are not sharp or scenes are complex including textures.

Local plane-fitting. Later in [18], authors presented a new model based on matching local plane fit over time. Fitted spatio-temporal planes overcome the sparsity of neighboring asynchronous events, allowing for more robust estimations. Instead of using temporal windows, the authors use the time from the events for the estimation, becoming also more biologically plausible. However, due to the eminently local nature of this approach, it does not correctly handle the aperture problem. To overcome this issue, [1] used a multi-scale spatial pooling approach. To do so, they initialized the optical flow using the local plane fitting approach. The authors then applied a multi-scale approach to estimate the refined optical flow, using a contrast-maximization approach being able to handle more complex scenes with general camera motion that involves 3D rotation and translation.

Variational. In [9], authors introduced another method to estimate dense optical flow, reconstructing intensity from events, using a variational approximation (based on Horn-Schunck [86]). The authors proposed to solve for the optical flow \mathbf{u} by reconstructing also the log intensity from the events L , by minimizing the Eq. 3.

$$\begin{aligned} \arg \min_{\mathbf{u}, L} & \int_{\Omega} \int_T \lambda_1 \|\mathbf{u}_x\|_1 + \lambda_2 \|\mathbf{u}_t\|_1 + \lambda_3 \|L_x\|_1 + \lambda_4 \|\langle L_x, \delta_t \mathbf{u} \rangle + L_t\|_1 + \\ & \lambda_5 h_C(L - L(t_p)) dt dx + \int_{\Omega} \sum_{i=2}^{|P(\mathbf{x})|} \|L(t_i) - L(t_{i-1}) - C p_i\|_1 dx \end{aligned} \quad (3)$$

The regularization terms are multiplied by λ_1 , λ_2 , λ_3 , and λ_5 , also the data term is multiplied by λ_4 (the OFC). The first three terms represent the variation of the optical flow and intensity, which are assumed to be smooth over the spatial neighborhood of each pixel Ω and over time T . The fifth term accounts for regions where no events are triggered, and adds up the difference between consecutive log intensity values $h_C(L(\mathbf{x}, t))$ if this difference is greater than the threshold C . Finally, the sixth term is the data term for regions with events, and it accounts for significant differences (greater than $|C p_i|$, with $p_i \in \{-1, 1\}$) being the polarity for event i in L . The authors mention that the work was still preliminary and needed further development to achieve accurate results.

Filter-banks. A differentiating approach was presented in [12], which proposed an OFC adapted to use phase gradient instead of intensity gradient by reconstructing the original intensity signal using the counts of events per pixel. Authors argue that this approach leads to a more stable estimation and to better deal with high-frequency textures. The model first separates textured edges and object contours and uses the local phase to estimate motion for the first case, employing the approach in [11] to estimate motion for the object contours. Both are innovative approaches but are limited to dealing with sequences with general motion and sharp edges. One of the problems with applying conventional optical flow methods to event processing is that event input is sparse and sometimes, active events are not enough to even form complete long edges. In [25], authors proposed a way of generating dense inputs from the sparse flow of events. They describe an approach that generates a grid-like representation based on the inverse exponential distance surface for each event, taking into account local neighborhoods. Then, they used a filter-bank-based algorithm to estimate dense optical flow.

Contrast maximization. Recently in [165], authors proposed a valuable multi-scale contrast-maximization model that reaches state-of-the-art performance estimating dense optical flow. In contrast-maximization methods, the objective is to recover the sharpness of edges from the projection of the flow of events on the image plane. The approach employs a multi-reference loss function to mitigate overfitting and a time-aware estimation technique to handle occlusions. Furthermore, the multi-scale strategy improves accuracy by avoiding local minima in the optimization. The experiments show that this approach is state-of-the-art for method-based and unsupervised approaches and, that is on par with learning-based supervised networks.

ANN-based approaches. ANNs are commonly used for event-based optical flow estimation, concretely using encoder-decoder architectures that learn hierarchical representations of motion from spatio-temporal relationships. The encoder compresses input patterns into a “feature vector” through convolutional or recurrent layers, while the decoder generates an output with the same size and shape as the input through deconvolutional or recurrent layers.

Supervised. These approaches require accurate ground truth, which is easier to achieve with simulators. Ground truth for real data requires additional hardware such as motion capture systems, that needs to be accurately synchronized with event sensors that have very high temporal resolution. Regarding the approaches, E-RAFT [72] is a state-of-the-art optical flow estimation model based on RAFT [174] since 2022. It uses two encoders to extract features from two consecutive event streams, builds an embedding, and estimates all-pairs correlations to iteratively produce optical flow. Recurrency is done via a GRU and a parallel context encoder that is fed into the next GRU using an additional module that warps the motion estimation from the previous temporal volumes into the next one.

Another interesting supervised learning approach for optical flow and depth estimation is EAGAN [113]. It is a generative adversarial architecture that uses a U-Net architecture with an attention-based transformer to learn temporal relationships between event sequences. EAGAN involves training both the generator and the discriminator simultaneously, using a loss function that is a combination of a supervised method for the discriminator, which requires the ground truth and the output optical flow from the generator, and a loss function similar to that of E-RAFT.

Self-Supervised. These are generally hybrid approaches: optical flow is estimated from event information while the supervision is done with grayscale images that are either simulated or captured using

Type	Subtype	Method	PE-based	DVS Resolution	Dataset	OF Estimation	Real Time	AEE
TM	Local plane-fitting	[18]	✓	346x260	NA	Sparse	Yes	NA
		[1]	✓	346x260	MVSEC	Sparse	Yes	1.59*
	Variational	[9]	✓	128x128	NA	Dense	No	NA
	Filter-banks	[12]	✓	NA	NA	Sparse	Yes	NA
		[25]	✓	NA	MVSEC	Dense	Yes	NA
	Contrast Maximization	[165]	✓	NA	MVSEC, DSEC	Dense	Yes	0.42*
ANN	Supervised	[72]	✓	346x260	MVSEC, DSEC	Dense	No	NA
		[113]	✓	346x260	MVSEC	Dense	Yes	0.212
	Self-Supervised	[207]	✗	346x260	MVSEC	Dense	No	0.49*
		[171]	✓	256x256	MVSEC	Dense	No	0.7*
	Unsupervised	[205]	✗	346x260	MVSEC	Sparse	No	0.32*
SNN		[108]	✓	346x260	MVSEC	Dense	NA	1.09*
	Self-Supervised	[80]	✓	NA	MVSEC	Dense	NA	0.45*
		[103]	✓	NA	MVSEC, DSEC	Dense	NA	0.44*
	Unsupervised	[78]	✓	304x240	NA	Dense	Yes	11%
		[109]	✗	256x256	MVSEC	Dense	NA	0.56*

Table 1: Most relevant optical flow algorithms using event cameras. PE refers to “Purely Event” based, NA refers to “Not Available”, and AEE stands for “Average End-point Error”. Regarding the results, the reported AAE is generally the average error for all the testing sequences in the dataset. Otherwise, * indicates that the reported AEE is for a particular sequence proposed in the work.

hybrid sensors such as DAVIS or stereo setups (conventional camera and event sensor). The loss function is a conventional photometric loss built by warping images using event-based optical flow estimation. However, bear in mind that the low temporal frequency of conventional cameras makes them more prone to blurring or light artifacts. EV-FlowNet [207] is an encoder-decoder inspired by U-Net and also, a baseline for optical flow estimation. Its loss function is the sum of the photometric loss and the smoothness loss that minimizes the optical flow variation. 3D-FlowNet [171] follows the same architecture but differs in that it uses 3D convolution layers to preserve temporal information. Learning requires both images and events, using a loss function similar to that of [207].

Unsupervised. These approaches are purely event-based and with a loss function that usually seeks the event alignment, for example, using motion compensation. The authors of EV-FlowNet [207], presented an approach using only events in [205]. In this work, the baseline architecture is used for the optical flow estimation, and the extended architecture with the pose model layers is used for depth and camera motion estimation. Authors use a loss function that relies on the sharpness of an image of warped events after the distortion of the event flow and its projection onto the image plane. A similar approach was followed in successive refinements of this work in [80, 142].

SNN-based approaches. SNN-based learning models have not reached their full potential yet and thus, they are not on par in accuracy with deep learning models.

Self-Supervised. In recent research, the integration of SNNs and ANNs has been explored to overcome the lack of backpropagation in SNNs. One such implementation is the Spike-FlowNet model, which combines SNN and ANN layers to create an encoder-decoder architecture. The encoder uses SNN layers, while the decoder is made up of ANN layers. This hybrid approach has been shown to save up to 17.6% of energy. The authors of [108] trained the ANN using the sum of two loss functions inspired by [207]. These are the photometric loss, which minimizes differences between two grayscale images, and the smoothness loss, which acts as a normalization term. The output is then passed into the SNN block that uses the surrogate gradient method [186] for training. In [80], authors used two different RNNs based on EV-FlowNet and FireNet architectures, along with various SNN alternatives for optical flow estimation. This work achieved good results in terms of accuracy but has high computational complexity. Recently, Adaptive-SpikeNet [103] aimed to adapt successful ANN models to fully-spiking models using SNNs. It combined the surrogate gradient method with self-supervised learning. The adapted models showed better efficiency than previous ANN and hybrid approaches, although ANNs still achieved lower error. In terms of energy consumption, SNN models were 5x more efficient.

Unsupervised. One of the first implementations in an embedded system was proposed in [78], with an SNN based on an adaptation of the Barlow & Levick model [10], using Direction Sensitive neurons. The network was implemented in the IBM TrueNorth Neurosynaptic System, achieving real-time operation with an energy consumption of 80 mW. Fusion-FlowNet was presented in [109] and it combines event-based and frame-based cameras to compute dense optical flow, in a model based on U-Net [155]. The

encoder uses SNN layers for the events and ANN convolutional layers for the frames, the residual blocks, and the decoder. SNN layers are endowed with signed activities, and learning is enabled via the surrogate gradient method to enable back-propagation.

Datasets. Collecting datasets is essential for learning-based methods but, it is also a very time-consuming task that requires ensuring diversity, completeness, high reliability, and accuracy.

MVSEC [204] builds on previous attempts [13, 20] that were limited due to their size or lack of diversity. The dataset was built using a stereo camera and an IMU, a pair of DAVIS neuromorphic sensors with integrated IMUs, a LiDAR sensor that provides ground truth, and a GPS unit. It contains 14 indoor and outdoor sequences including day and night scenes, relevant to assess the high-dynamic range of the neuromorphic sensors.

DSEC overcomes some of the limitations in MVSEC [75] as scenes that lack some ground truth or source data, or poor 3D data for some scenes due to their stereo-rig baseline distance compared to the maximum depth of these scenes. In this case, the authors use two Propesee Gen3.1 neuromorphic sensors, a stereo pair of cameras, a LiDAR, and a GPS sensor. All 53 sequences in this dataset were collected from a car, driving in urban, suburban, and rural areas, including day and night sequences, with some challenging artifacts such as a car driving out of a tunnel. Table 3 summarizes the most relevant features of both datasets.

4.1.2 Ego-Motion

The biggest challenge for autonomous robotic systems is to execute fast and secure trajectories while navigating. To do it, their relative pose with respect to the scene needs to be accurately estimated. For the last few years, precise navigation solutions [59, 154] have demonstrated that current control algorithms for aggressive maneuvers are robust enough. However, all these precise solutions are not computed from the robot perception subsystem estimating their own 3D self-motion or ego-motion. Most of these solutions only work on controlled scenarios and use motion-capture (MoCap) systems, which have been widely proven to be much faster and more reliable than any other solution for real-time pose estimation. However, these are expensive systems that need active cameras and can only be used indoors.

The set of methods that allow estimating the robot motion (translation and rotation) relative to a reference frame is known as visual odometry (VO) or ego-motion. Efficient onboard solutions with monocular cameras, stereo cameras, or laser [91, 115] have been proven sufficient for motion estimation. However, their latency still limits the processing pipeline in dynamic environments or at high speeds. Brain-inspired solutions are good candidates for onboard processing due to their low computational complexity and energy consumption. Also, these solutions leverage the agility of the robot by minimizing the latency of perception which is a crucial constraint to achieving real-time navigation at high speeds [56]. Notice that some methods in Section 4.1.1 also address the ego-motion estimation problem usually as a by-product when estimating optical flow. To avoid redundancy, this section focuses only on ego-motion estimation as shown in Table 2. The presented works are divided into monocular and stereo approaches, and categorized attending to the event stream data representation.

Monocular Ego-Motion. Most pose estimation methods rely on a single camera, avoiding the need to solve the data association problem of multi-view camera systems. Monocular approaches have been shown to be sufficient to estimate the ego-motion.

Event Accumulation. The first onboard low-latency approach for 6 degrees of freedom estimation for aggressive maneuvers using an event camera was presented in [134]. In this approach, lines are built from sparse events and for each new event, the method decides which close-by lines are to be updated. The pose estimation is computed by minimizing the sum of squared distances between the new events and the reprojection of each line, that belongs to a known pattern. The method estimates the pose under very challenging conditions but requires this pattern.

In [99], the authors describe a real-time joint estimation of 6-DoF camera motion, 3D structure, and intensity using three decoupled EKF-based probabilistic filters. The proposed approach shows good results in unstructured environments with no prior knowledge of the scene. EVO [147] is a purely geometric event-based tracking and mapping method that runs on the CPU in real time and outputs up to several hundred pose estimates per second. It involves running a 3D reconstruction algorithm [149], in parallel with an event-based image-to-model alignment for tracking. In frame-based direct methods, this alignment between two frames is computed as a minimization of its photometric error [60]. However, since event cameras do only respond to edges in the scene, this photometric error is replaced by a geometric alignment between two edge images as shown in Eq. 4.

<i>Event Representation</i>	<i>Method</i>	<i>PE-based</i>	<i>Sensor</i>	<i>Resolution</i>	<i>Dataset</i>	<i>Accuracy</i>
Event Accumulation	[134]	✓	Monocular	128 x 128	Custom	Trans. error: 0.108 ± 0.078 m Orient. error: $5.1 \pm 2.4^\circ$
	[99]	✓	Monocular	NA	Custom	Qualitative Results Only
	[147]	✓	Monocular	240 x 180	Custom	Trans. error: $\leq 0.2\%$ Orient. error: $\leq 3^\circ$
	[156]	✗	Monocular Events + Frames + IMU	240 x 180	Event Camera Dataset	Trans. error: $\leq 0.24\%$ Orient. error: $\leq 0.06^\circ$
	[136]	✓	Monocular	240 x 180	Event Camera Dataset	Trans. error: 0.036 m Orient. error: 2.341°
	[39]	✓	Monocular	240 x 180	Custom	Trans. error: < 5% Orient. error: < 4°
Probabilistic	[31]	✗	Monocular Events + Frames	128 x 128	Custom	Trans. error: NA Orient. error: $\leq 1^\circ$
	[62]	✗	Monocular Events + RGBD Map	128 x 128	Custom	Trans. error: $\leq 1.8\%$ Orient. error: $\leq 1.04^\circ$
Events Volume	[206]	✗	Monocular Events + IMU	240 x 180	Event Camera Dataset	Trans. error: $\leq 2.57\%$ Orient. error: $\leq 0.27^\circ/m$
Time Surfaces	[203]	✓	Stereo	346 x 260	MVSEC * Event Camera Dataset *	Trans. error: ≤ 0.053 m/s Orient. error: $\leq 1.93^\circ/s$
	[77]	✓	Stereo	346 x 260	MVSEC * DSEC *	Trans. error: $\leq 8.76\%$ Orient. error: $\leq 1.17^\circ/m$
	[208]	✓	Stereo Events + RGBD	640x480	Custom	Trans. error: ≤ 0.0142 m/s Orient. error: $\leq 1.57^\circ/s$
ANN Unsupervised	[194]	✓	Monocular	NA	MVSEC *	Trans. error: 3.65° Orient. error: 0.23°

Table 2: Most relevant ego-motion algorithms using event cameras. PE refers to “Purely Event” based, and NA stands for “Not Available”. The accuracy values are given as the mean of the individually presented results of every sequence from the dataset used. * stands for not using all the sequences of the dataset.

$$\arg \min_{\Delta P} \sum_{\mathbf{x}} (M(\mathcal{W}(\mathbf{x}, \Delta P)) - \Gamma(\mathcal{W}(\mathbf{x}, P)))^2 \quad (4)$$

That is indeed the inverse compositional Lucas-Kanade alignment algorithm [6], where M corresponds to the *template* of the projected semi-dense map that is incrementally built, which is used to compare with the current event accumulated image Γ . To achieve this, a wrap-rigid-body transformation \mathcal{W} , at a point \mathbf{x} in the image plane of the template M , is minimized to compute the pose estimate P (Eq. 4). It is a minimization for the incremental pose estimation ΔP that is, the complete set of transformations of the map from the origin to the current pose. Since transformations are relative, this method is prone to incremental error accumulation.

UltimateSLAM is a holistic approach [156] which consists of a pipeline for robust and accurate state estimation that tightly combines events, frames, and inertial measurements. The main idea is to build event frames by aggregating spatiotemporal windows of events, and then use the FAST corner detector to extract features that will be tracked using the Lucas-Kanade tracker [117]. The camera pose is updated fusing conventional frames and virtual motion-compensated event frames [148], with a final refinement fusing inertial cues. This final problem is solved as a joint optimization whose cost function J shown in Eq. 5 consists of two reprojection errors e^i of the observations from the event-based ($i = 0$) and the frame-based ($i = 1$) cameras, plus the inertial error e_s .

$$J = \sum_{i=0}^1 \sum_{k=1}^K \sum_{j \in \mathcal{J}(i,k)} e^{i,j,k^T} W_r^{i,j,k} e^{i,j,k} + \sum_{k=1}^{K-1} e_s^{k^T} W_s^k e_s^k \quad (5)$$

In Eq. 5, k is the frame index, j the landmark index, and $\mathcal{J}(i,k)$ is the set of landmark indices for each sensor i and frame k . The terms $W_r^{i,j,k}$ and W_s^k are the information matrices of the landmarks and inertial measurements respectively. Their results show the complementarity of conventional frames and events, boosting accuracy in 130% compared to using only events and IMU data.

[136] proposed a Stacked Spatial LSTM Network (SP-LSTM) to predict the camera pose by taking just event-wrapped images as input. In this work, features from the event images are extracted using a CNN. The output of the network is reshaped and used as input for the SP-LSTM module. This network aims to learn spatial dependencies in the feature space, which can improve the relocalization of the camera pose. This network has an inference time of 5ms, which can be considered a real-time system but is not enough for high-speed robotic applications. A 6-DoF implementation relying on an EKF for pose estimation was proposed in [39]. The key of the algorithm is the proposed event-to-map

association. First, a 3D point cloud is initialized by event projection to the parallel image plane at a fixed distance, assuming that the sensor is initially static. Then, a pixel-to-3D-point association is performed and subsequently tracked for pose estimation with an EKF, continuously updating the map to track the event-to-map association. The main advantage of this event-by-event method is that it updates the overall estimation with every new incoming event and thus, it greatly reduces latency.

Probabilistic. [31] solves visual odometry using an event sensor and a conventional camera. The solution is limited, and only rotations are accurately computed while translation estimation is very noisy. A differential approximation is described in [62], in which authors tackle full-motion estimation for augmented/virtual reality (AR/VR) purposes, using a single event camera, and a photometric depth map built with a traditional dense 3D reconstruction and conventional images. Although the estimation is accurate, the approach requires RGB-D sensors or an additional method to estimate depth from conventional frames.

Event Volume. EVIO [206] represents the first event-based feature tracker algorithm to fuse event data streams with an inertial measurement unit. Features are detected within the orthogonal projection of the events onto the image plane, using the FAST corner detector [157]. Finally, it fuses IMU data with EKF to improve the tracking accuracy.

ANN Unsupervised. The authors of [194], presented an approach purely based on events with an Evenly-Cascaded convolutional Network (ECN), using an encoder-decoder architecture to estimate the depth and a second neural network that takes consecutive slices of events to estimate the camera pose. Then, these two neural networks are used to estimate the optical flow.

Stereo Ego-Motion. There are also stereo implementations that try to mimic the dual-eye capabilities of biological systems and exploit stereo advantages for depth estimation (a very detailed survey on stereo event-based vision can be found in [169]).

Time Surfaces. ESVO [203] is the first parallel tracking-and-mapping approach using a stereo event camera. The mapping module builds a semidense 3D map of the scene by fusing depth estimates from multiple viewpoints (obtained by spatio-temporal consistency) in a probabilistic fashion. Tracking of event features is performed by fusing the map and the stream of events. Event features are extracted from spatio-temporal surfaces also known as time surfaces $\mathcal{S}(\mathbf{x}, t) = e^{-(t-t_{last})/\tau}$ [106] which encode the motion history of the edges. The time surface \mathcal{S} is a 2D map that is updated according to the timestamp of the last event t_{last} occurring at a pixel location \mathbf{x} and a decay rate parameter τ . Looking at time surfaces, large values show the locations where the current edges are. The estimation is done by aligning time surfaces over time using the depth maps, as if time surfaces were artificial images. In fact, the tracking method consists of aligning these regions with the computed depth map once it is warped to the time surface frame by a candidate pose. Thus, to solve the compositional Lucas-Kanade method, authors compute the motion of the camera as an optimization with the cost function shown in Eq. 6.

$$F(\Delta\theta) = \sum_{\mathbf{x} \in S^{F_{ref}}} (\mathcal{S}(\mathcal{W}(\mathcal{W}(\mathbf{x}, \delta; \Delta\theta); \theta), k))^2 \quad (6)$$

Where θ stands for the motion (orientation and translation), $S^{F_{ref}}$ is the set of pixels \mathbf{x} in the reference frame F_{ref} with a valid inverse depth z , and \mathcal{W} is a wrapping function which aligns the semi-dense map with the current time surface at time k .

More recently, [77] proposed an event stereo visual odometry method that relies on features that are matched through consecutive left and right event time surfaces. The method does not produce pose information at a fixed frequency, but dynamically, and it only depends on the number of incoming events to ensure sufficient visual information for reliable feature detection. Finally, [208] presented a hybrid stereo setup with a depth camera and an event camera. This approach first extracts edges from time surface maps then assigns depth to them using an RGB-D camera, and finally registers subsequent views by 3D-2D edge alignment. Their results outperform ESVO and are robust under challenging illumination conditions. However, the requirement of the depth camera makes it biologically unfeasible.

Datasets. A first attempt for a small dataset was done in [13], including several sequences of different general motions on unstructured scenes. The dataset contains optical flow, depth maps, and 3D motion ground truth data. As for large-scale datasets, the main efforts have been *MVSEC* and *DSEC* datasets, already described in Section 4.1.1.

TUM-VIE [100] contains handheld and head-mounted recordings mostly indoors. It also includes some sequences with rapid motion during sports practice with changing illumination conditions. It was recorded ensuring hardware synchronization, using a stereo pair of conventional cameras, two Prophesee Gen4 event sensors, an inertial sensor, and the MoCap system at 120 Hz.

Finally, the *VECtor* dataset [66] includes two sets of sequences: 1) 12 small-scale indoor sequences, some of them recorded under low illumination conditions; 2) 6 large-scale sequences recorded while

Dataset	Sensor	Resolution	Baseline (cm)	FOV (°)	Illumination	Max. depth (m)	Total
MVSEC [204]	Davis346	346x260	10	74.8	Indoor/outdoor Day/night	(Indoor) 8.4	14 sequences (4083 s)
DSEC [71]	2 x Prophesee Gen3	640x480	60	60.1	Outdoor Day/night	50	53 sequences (3193 s)
TUM-VIE [100]	2 x Prophesee Gen4	1280x720	11.84	90	Indoor/outdoor Artifacts, day/night	NA	21 sequences (2880 s)
VECtor [66]	2 x Prophesee Gen3	640x480	17	82	Indoor/outdoor Day/night	5 (indoor)	18 sequences
EV-IMO [129]	Davis	346x260	NA	80	Indoor	NA	6 sets containing 19 sequences (1920 s)
MOD and MOD++ [161] and [141]	None	346x260	NA	NA	Outdoor Day	NA	6 (MOD) + 2 (MOD++) sequences
EV-IMO2 [26]	2 x Prophesee Gen3 Samsung DVS	640x480	22	75	Indoor low light	NA	173 sequences (2460 s)

Table 3: Summary of parameters of the most popular event-based datasets. NA stands for “Not Available”.

walking or driving a scooter. The setup includes a stereo pair of Prophesee Gen3, a stereo pair of conventional cameras, an RGB-D camera, an IMU, a LiDAR, a Laser Scanner, and a MoCap system that provides ground truth. All datasets are summarized in Table 3.

4.2 Mapping

When deploying a robot in an unknown scenario the map might not be available. To achieve truly autonomous capabilities, the robot must be able to build this map in real-time while it moves. First, the robot estimates depth and then, it builds the map as an aggregation of sparse 3D features. This section describes depth estimation methods and common mapping representations with events.

4.2.1 Depth Estimation and 3D Reconstruction

Autonomous navigation in any environment requires a 3D representation of the scene to plan the trajectory to follow or simply, to avoid collisions. Depth cues are crucial for mapping or planning algorithms, especially if environments are dynamic, or in the most simple case, for reactive controllers. Experiments such as [88], prove that edges are more relevant than color to estimating depth. Therefore, as event cameras do only respond to salient edges and textures, they are the best candidates for accurate depth estimation of the scene. In this section, we present stereo and monocular event-based solutions.

Monocular Depth Estimation. Geometrical-based approaches using a unique camera for 3D semi-dense depth reconstruction of the space require information about the moving camera, which means using its ego-motion as input. Thus, a direct point cloud generation cannot be achieved.

EMVS [149] was the first multi-view stereo approach that reconstructed the 3D structure of the scene in real-time using a single event camera. However, in this approach, the model obtained the camera pose from an external tracking system. In [63], the authors presented a unifying framework to simultaneously solve the estimation of 3D motion, depth, and optical flow. The method looks for point trajectories on the image plane that best fit the event data by reconstructing the scene dynamics. Additionally, the framework outputs motion-corrected event images using an objective function to measure how well the events are aligned along the candidate trajectories using contrast maximization. Finally, [64] provides a comprehensive list of objective/focus-loss functions for motion compensation approaches.

[85] proposed the first system for dense depth monocular estimation using only events with a recurrent convolution network. This is done to leverage the temporal consistency of the event stream. The proposed methodology shows better accuracy results than existing methods. The loss function, Eq. 7, is the weighted sum of the scale-invariant gradient matching loss $\mathcal{L}_{k,si}$ and the multi-scale-invariant gradient matching loss $\mathcal{L}_{k,grad}$. This last term allows the network to predict smooth depth maps, enforcing sharp depth discontinuities.

$$\begin{aligned} \mathcal{L}_{k,si} &= \frac{1}{n} \sum_{\mathbf{x}} (R_k(\mathbf{x}))^2 - \frac{1}{n^2} \left(\sum_{\mathbf{x}} R_k(\mathbf{x}) \right)^2 \\ \mathcal{L}_{k,grad} &= \frac{1}{n} \sum_s \sum_{\mathbf{x}} \|\nabla_x R_k^s(\mathbf{x})\|_1 + \|\nabla_y R_k^s(\mathbf{x})\|_1 \end{aligned} \quad (7)$$

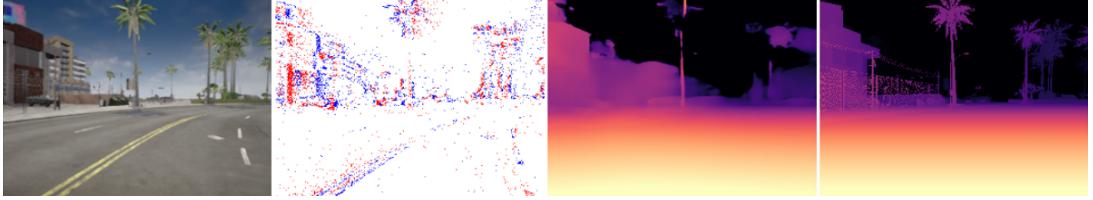


Figure 5: Qualitative results comparing depth prediction (extracted from [85]). From left to right: frames, event stream, event-based predicted depth, and ground-truth depth.

Where R_k and R_k^s are the residual of the depth, and that same residual at scale s respectively, for n valid ground-truth pixels \mathbf{x} (more precisely, for a sequence of ground-truth depth maps).

A learning-based monocular depth estimation was proposed by [70] using events and frames to capture richer information from the scene. Due to the architectural limitation of traditional Recurrent Neural Networks (RNN) [167] for processing asynchronous events and synchronous frames, authors propose a Recurrent Asynchronous Multi-modal (RAM) network. This network enables the processing of any irregular data from multimodal sensors. The main drawback of this approach is the need to use both frames and event streams. Finally, in [79], authors presented a neuromorphic approach employing an SNN that uses depth from defocus (DFD), proposing a hardware-friendly implementation that estimates sparse depth.

Stereo Depth Estimation. Most stereo depth works to tackle the problem known as *instantaneous* stereo depth estimation. This means generating a 3D point cloud in a very short time (ideally, event by event) using a stereo pair and ensuring clock synchronization. Synchronization is limited due to factors such as different pixel sensitivities that can induce different activation/deactivation timestamps, or sensor bus transmission congestion that will delay the signal. The 3D reconstruction general pipeline follows a two-step process: 1) solving the stereo event matching correspondence (which is the most computationally intensive step), for example, using asynchronous epipolar constraint as shown in [16, 153]; 2) computing each 3D point by triangulation following classical solutions [82].

[144] estimated depth using a dynamic cooperative network for asynchronous stereo matching. The key aspect of this disparity calculation approach is that it does not only consider the temporal similarity of the events but also the spatiotemporal neighborhood. The first dense depth map reconstruction was presented in [49]. The work proposed a discriminative event feature descriptor, which showed invariance to changes in translation, rotation, and scale. It also proposed an event-enhancing method that is crucial for finding matches between the stereo event streams. More recently, authors in [202] proposed a non-instantaneous 3D reconstruction algorithm with a stereo event-based configuration by matching temporal coherence of time-surface maps [106] from left and right event sensors. Depth estimate is solved via energy minimization exploiting the spatiotemporal consistency, instead of the classical event matching plus triangulation pipeline.

In [151], authors proposed an instantaneous stereo depth estimation approach by using SNNs with events as input. The intention of using SNNs for the stereo-matching computation is to reduce the intrinsic bottleneck of this process. This architecture is implemented on a mixed-signal neuromorphic processor, leveraging this to a more similar brain-inspired solution. Finally, the work in [146] uses a hardware-friendly fully-spiking SNN that is fed with two trains of events from different DVS neuromorphic sensors, using an encoder-decoder architecture. Interestingly, this network achieves state-of-the-art accuracy, outperforming its image-based equivalent ANN. A thorough review of depth estimation can be found in [61].

Datasets. There have been different attempts to provide datasets that include structured and non-structured scenarios, mostly indoors due to the limitations of RGB-D cameras that were used to collect ground truth depth. However, nowadays the most popular datasets for depth estimation are *MVSEC* already described in Section 4.1.1, and *TUM-VIE* described in Section 4.1.2.

4.2.2 Mapping Representation

One of the most important stages in the pipeline of autonomous navigation is the representation of the environment, enabling the robot to plan its routes. Most works in the literature consider two representations: metric and topological maps.

Metric Maps. They consider geometric information of the environment and are built to establish a fixed reference frame. Solutions range from sparse landmark-based and occupancy grid maps to 3D semi-dense or fully-dense reconstructions. Occupancy grids are the most popular kind of metric maps

[87, 139] due to their good localization accuracy vs. computational complexity trade-off. These maps are represented as an array of 2D pixels or 3D voxels that encode occupancy probability.

Topological Maps. Instead of considering the geometric relationship between landmarks, these maps gather adjacency information. Their most common representation are graphs where each node encodes a place or region of interest, and links that represent the adjacency information of the connection between nodes [22]. This representation reduces the computational complexity of planning algorithms since they do not need to find a full path in the geometrical space but a sequence of places to visit in a graph [118, 188].

4.3 Simultaneous Localization and Mapping - SLAM

Although ego-motion methods have proven to be sufficient for pose estimation, their use for continuous localization does not solve the drift and error accumulation problems for long trajectories. Moreover, fine mapping is not possible without a continuous and robust pose estimation. While a robot navigates around an unexplored environment, the relative positions of the observed landmarks are all correlated due to the error of the robot's estimated location. This would make the problem intractable since for every new landmark the robot would need a new state with the robot position and the relationship of this new landmark to all the previous ones. It was not until 1995, with the new works on the convergence of the combined estimation of mapping and localization [42], that authors realized that the optimization was feasible. A large number of landmarks is actually crucial to obtaining accurate solutions.

Regarding localization, some approaches mainly focus on incrementally estimating the path of the agent, pose by pose, while performing local optimization. In the case of Simultaneously Localization and Mapping (SLAM), it aims at obtaining a globally consistent estimate of the agent trajectory while simultaneously mapping its surroundings [196], reducing drift errors. This is achieved by recognizing a location that has been previously visited and it is known as *loop closure*. In other words, loop closure is the primary rectification step responsible for reducing the pose estimation drift errors when performing long trajectories, ensuring global consistency [178]. Due to the sparse nature of the event streams, event-based approaches have the potential to achieve a significant reduction in computational complexity compared to current visual SLAM algorithms using conventional cameras.

Next, the most relevant brain-inspired SLAM approaches are presented. In [184], the authors presented vSLAM, an event-based 2D-SLAM method, as an extension of a previous event-based tracking formulation that adapts a particle filter condensation algorithm for self-localization [183]. With every new event, the tracking algorithm uses a set of particles for a multivariate probabilistic estimate of the current system state. Each particle conveys the corresponding state estimation and its likelihood score. The map for tracking features is dynamically updated during localization and built as an occupancy map with each pixel representing the likelihood of events being triggered at its coordinates. Although the method reaches good accuracy, the solution is limited to 2D navigation and high-contrast scenarios. Moreover, there is no component for global consistency loop-closure. Lately, the same authors presented another extension for 3D-SLAM, but this time using a D-eDVS, a combined event-based-3D sensor (DVS + RGB-D) [185]. Although this approach presented good results for 3D scenarios, the combination with the depth sensor limits the latency of the event camera. Again, a global optimizer to handle loop closure is still not integrated.

The first truly event-based SLAM system ensuring loop closure was proposed by [125]. It is considered a breakthrough in adapting current algorithms to work with event cameras. This naive approach consists of three key modules: 1) a loop closure algorithm built over the results of SeqSLAM [127]; 2) a limited visual odometry method constrained to the ground plane; 3) a semi-metric, topologically correct mapping tool based on the experience mapping algorithm [126]. For landmark recognition, SeqSLAM needs frame sequences. To adapt it for an event camera, events are temporally accumulated to create synthetic images. Finally, loop closure is integrated with the localization in the experience map as explained in [122], showing reliable place recognition in diverse indoor and outdoor scenarios and navigating at different speeds. Authors in [137] proposed ASL-SLAM, an event-based visual SLAM approach that tracks features and event lines simultaneously. This approach showed improvement especially when operating under the presence of motion blur or low-frequency textures and under different lighting conditions. Its pipeline is similar to the PL-SLAM [145], but uses synchronously extracted event lines combined with frame-based point feature correspondences. Following the same idea, in [32] authors built an approach that is based on the minimization of the event-line reprojection error. The method uses an error-state Kalman filter formulated entirely with Lie theory. Again, none of these approaches implement global loop closure. Finally, in [182] authors proposed a method for an asynchronous Kalman

Filter using an event sensor. The method is demonstrated in several applications including the deblurring of conventional frames when aggressive motion occurs. Recently, [58] presented an event-based visual place recognition (VPR) algorithm that, given a new observation, retrieves the best matching reference from a database. This event-based VPR represents a valuable step towards loop closure on event-based SLAM algorithms.

4.4 Planning and actuation

In robotics, planning methods split any high-level task into a sequence of low-level specific actions to accomplish it. For navigation, these actions are mainly the movements that the robot needs to perform to reach the final goal of the task. Most methods consider three areas in robotics planning: task planning [92] that generates a sequence of actions; motion planning [33] that generates a sequence of movements; and trajectory/path planning [164] that produces a trajectory to reach the goal. For a detailed analysis of path planning methods, we encourage the reader to check [95, 96].

For the sake of completeness, this section briefly introduces the most popular path-planning approaches that are also the baselines for most bio-inspired works in the literature.

4.4.1 Path planning: local and global

Path planning algorithms are used in autonomous robotic navigation to analyze and create trajectories from the robot’s current location, or *origin*, to the destination, which is referred to as *goal*. Depending on the considered criteria, these path planning approaches optimize qualities such as traversed distance, time, battery level, or velocity, to achieve safe, collision-free, or efficient and least-cost paths.

Planning considers local and global strategies. In fact a robust autonomous robot integrates both. *Global planning* consists of finding a set of way-points that, when linked, describe a unique trajectory between the origin and the goal of the mission. Each way-point encodes a position-configuration change (maneuvers) or specific high-level action (such as taking a photo or measuring a magnitude). These planners normally do not take into account the robot’s kinematic model. In contrast, *Local planning* finds a trajectory between two way-points, avoiding obstacles. Global planning is done offline before any movement is executed and thus, it is risky since the current map may be imperfect or incomplete. Local mapping reduces risks locally, for example by updating the map with unexpected information or avoiding dynamic obstacles. Next, the two most well-known planners in the field are briefly described.

Rapidly-Exploring Random Trees (RRT). This algorithm [107] is an incremental sampling-based planner that randomly samples and analyzes the map until a connection between origin and goal is found. This feature reduces computing time when planning paths, compared to other approaches that look for them in dense space. The algorithm works as follows: 1) The origin of the robot is set as the *seed* of the *tree*, which is a graph. 2) A random configuration (position) in the space is sampled and then, a collision detection algorithm validates it and checks for a free connection to the closest point of the current tree. 3) This process is iterated until a new node is close enough to the desired goal, outputting the path from the origin to this last node. Although it always guarantees at least one possible path in a limited time, sampling is random, and therefore, the path is sub-optimal. Finally, RRT can be used as a global or local planner. Some variations include RRT* that finds optimal paths [95]. However, RRT* is computationally expensive, which makes it less suitable for onboard execution. FAST-RRT* [138] is a hybrid pseudo-optimal solution that combines RRT and RRT*, finding a more optimal path than RRT and faster than RRT*. The method uses a local-neighborhood tree reconnection of the path using a cost-based function.

Probabilistic RoadMaps (PRM). In this case, the planner builds a graph of the map that will be used for future plans [98]. In contrast to RRT which has to sample and analyze the whole map for every requested plan, the PRM algorithm only needs to do it once for the entire exploration. When the map is sampled as a graph-like representation, better known as *roadmap*, paths are found using graph-searching algorithms that considerably save computing resources. A roadmap is a connected graph in which nodes correspond to robot configurations (ie. positions in the space), and edges are simple trajectories between these nodes. The algorithm works as follows: 1) The configuration space is probabilistically sampled to create free nodes, providing a rather uniform covering of the space. 2) For each new node, a few neighbors in the graph are selected, and a local planner tries to connect them. 3) When this *construction* phase is over, a second *expansion* step is executed to refine the current graph and ensure its connectivity. 4) When a path is requested, two more nodes are added to the roadmap, if possible: the origin and the goal. Finally, the path is discovered by a graph-search algorithm. Popular graph-searching algorithms

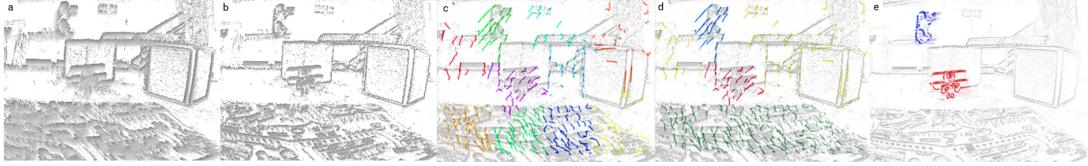


Figure 6: Overview of the proposed pipeline in [141]. (a) Projection of the raw event cloud without motion compensation, (b) Projection of event cloud after global motion compensation, (c) Sparse tracklets extracted on compensated event cloud, (d) Merged feature clusters based on contrast and distance metrics, (e) Output of the pipeline is the cluster of events. The cluster membership is color-coded where the gray color indicates the background cluster (extracted from [141]).

like Dijkstra , A* can be used to find these paths for the resulting roadmap. Some popular variants of this algorithm are PRM* [95] and Lazy-PRM [23].

4.4.2 IMO Detection

When the robot navigates the space, all points in the image are with respect to the perspective of the robot. Changes in the image can be due to the movement of the camera itself (ego-motion) or to the actual movement of objects in the scene (IMOs). **The goal of IMO detection is to segment moving objects in the sequence while the robot is exploring.** This can be difficult to accomplish because it is hard to estimate when the whole scene seems to be moving, due to the self-motion. Additionally, IMO detection usually involves the estimation of the pose and velocity of these dynamic objects. **Finally, IMO segmentation approaches can be categorized [120] into those focusing on the segmentation of a single object that is moving in the scene , and those that perform motion segmentation for multiple objects with a moving camera.**

There are several ways to segment independently moving objects. Early approaches in the literature tried to subtract the overall camera motion from the estimated optical flow to segment the object. These techniques, in brain-inspired literature, are commonly referred to as ***motion compensation***. Then, more sophisticated approaches use ego-motion, projecting the 3D motion in the scene into the 2D camera plane, and iteratively subtracting it from the estimated motion to segment areas with residual motion until convergence. Finally, most recent approaches directly apply Deep Learning techniques using supervised and unsupervised methods.

In [128], authors proposed a computational graphic approach on the 3D point cloud of spatiotemporal events for ego-motion estimation. In their real-time pipeline, time windows are selected and then a motion compensation algorithm is applied to estimate the ego-motion of the camera, by fitting a **parametric motion model**. After ego-motion estimation, method continues with the segmentation of IMOs by clustering for the motion residuals. **0-MMS** is an approach that also takes advantage of monocular motion compensation, and was presented in [141]. The authors extracted event-based features and applied them for motion compensation while tracking them, yielding an approach that can accurately segment multiple independent moving objects in the scene, by following a strategy based on **motion clusters**.

Authors in [3] presented a CNN encoder-decoder for semantic segmentation using event cameras, with an encoder inspired by the *Xception model* [38] and a light decoder. The work compares different alternatives and proves the advantage of hybrid approaches that combine images and events. In [129] the authors used two CNNs based on [194]. The first neural network was devoted to estimating the depth of the scene by means of an event window. The second neural network was designed to estimate the position and orientation of each pixel using several event windows. Finally, by combining these outputs, optical flow and segmented IMOs are obtained. A key of this study is that they trained the shadow network with just 40k parameters and introduced a new dataset called EV-IMO for IMOs. Also, they mentioned that this architecture can generalize in a variety of scenes.

Finally, there is also an interesting method that applies SNNs to the problem of segmenting IMOs. SpikeMS [140] is a fully end-to-end SNN for motion segmentation that employs an encoder-decoder architecture based on [130], but feeding the events directly into the network without the need to use synchronized temporal windows. They obtained a good result but without remarkable improvement over those obtained with other methods.

Datasets. The most widely used datasets for IMO detection are EV-IMO [129] and EV-IMO2 [26]. However, previous attempts also achieved great popularity MOD [161] and MOD++ [141].

MOD was originally intended for dynamic obstacle avoidance, attempting to avoid the pitfalls of other datasets that were built assuming static scenes [204]. It contains 6 synthetic sequences that represent

real-world indoor and outdoor scenarios. Every sequence has several light sources to recreate different illumination conditions with the camera moving around. The objects in the scene have been chosen using random textures with different shapes such as a ball, a drone, or a car. MOD++ extends MOD with additional sequences. *EV-IMO2* extends EV-IMO with more complex scenarios, including RGB images, and increasing the number of sensors providing data with 2 Prophesee Gen3, a Samsung Gen3, and a conventional Flea3 camera, and a MoCap system to obtain the ground truth. The dataset contains 173 indoor sequences grouped into three categories: independently moving household objects, static scenes, and basic motions in shallow scenes. Some of them were recorded in low light conditions. There are over 20 different objects such as small drones, toy blocks, or remote control cars. These datasets are summarized in Table 3.

4.4.3 Obstacle avoidance

When navigating in any environment, the robot needs to reach the destination by dodging any unexpected static or dynamic obstacle. This task is particularly crucial when the robot navigates a dynamic environment and needs to act as fast as possible to avoid collision with a static obstacle or even another mobile robot. Challenges for obstacle avoidance include cluttered environments or fast objects that require very precise and high-speed maneuvers. Thus, reducing the perception latency for high-speed obstacle avoidance has become a challenge in autonomous robotics [14, 159]. Precisely, event-based sensors have become the best candidates for obstacle avoidance given their very low latency.

[55] presented a study analyzing, from a theoretical perspective, the perception latency vs. actuation agility trade-off at maximum speed, to navigate autonomously in a static scenario. In their experiments, an event camera is used for dynamic obstacle avoidance on a quadrotor platform. Event cameras allow the robot to achieve a speed increase between 7% and 12% when compared with conventional cameras. EVDodgeNet [161] is a deep learning network that performs multiple dynamic obstacle avoidance using event cameras. The method uses the inertial inputs from the platform and the event stream, which is stabilized using motion-compensated by using a neural network. The stabilized input is then processed by two additional networks that perform ego-motion and IMO detection estimation. All the results are used as a control policy for dodging multiple dynamic obstacles. Another interesting application for obstacle avoidance in low lighting conditions using event cameras was proposed in [193]. The pipeline has four stages: 1) background event activity noise cancellation; 2) object detection module using an adaptive slicing algorithm and a Hough transform [190] and in parallel, a module that performs 2D-corner detection [181]; 3) depth estimation of the 2D features; 4) estimation of the relative velocity of the object to avoid collision.

4.4.4 Tracking

Tracking an object starts with its detection is why some works referenced in the IMO section also include tracking. Here, we will discuss works focused solely on tracking.

Tracking aims to detect and follow an object during the action and is needed in planning and obstacle avoidance. But it can also be used for object tracking alone. Most tracking approaches are mainly linked to the detection of IMOs, their classification, and labeling for a certain time, depending on the required task. Although these approaches are computationally intensive, there are implementations with neural networks that run in real time thanks to event cameras. For example, [198] proposes a spiking transformer network. The architecture is based on SNN and uses two components. The first is the SNNformer Feature Extractor (SFE), which takes advantage of the Swin transformers [116] to extract global spatial features from events. The second component is temporal-spatial feature fusion (TSFF). This method runs in real-time outperforming the state-of-the-art but, it only tracks a single object.

5 Spatial-cell Models for Cognitive Autonomous Navigation

Apart from the models for isolated components, holistic approaches for autonomous navigation that mimic different brain areas and structures involved in navigation have been proposed. The goal is to take advantage of neuroscientific developments to formulate new effective lightweight solutions deployable in small and low-power computing platforms.

As mentioned in Section 2.3, although other brain areas do play a role in navigation, the entorhinal cortex and the hippocampus are the main ones in charge of building a cognitive map of the scene and supporting localization within it. Simplifying the roles of the specialized cells that are localized

in these brain regions: PCs could be seen as spatial units for mapping external features onto internal and local metric coordinates in a distributed cognitive map; HDCs could be conceived as some kind of internal compass; and GCs represent the position within this map. Furthermore, space representation is a process done in an idiothetic manner. Over time, these brain areas integrate self-motion cues from different sensory inputs to estimate pose and location relative to a reference initial point, the so-called path integration. Next, we introduce computational models for the most relevant autonomous navigation works that mimic the working principles of these brain regions.

5.1 Computational models for spatial cells

Different computational models of the PCs firing rate have been proposed based on the results of biological experiments on rats. [83] modeled the inputs to the PCs as a population of Boundary Vector Cells (BVC), each of which fires maximally when a boundary is at a particular distance and allocentric direction to the agent [15, 110]. Thus, the PCs firing rate is modeled as proportional to the thresholded sum of the BVC inputs that it receives by using a threshold-linear transfer function. The model is able to describe PCs adaptation during learning, such as the consolidation of place fields and removal of duplicate fields. However, it does not perform well when map cues are removed, missing the remapping feature of PCs. More recently, [119] introduced a detailed morphological and biophysical model implementation for the creation of PCs.

Regarding GCs and HDCs, for example, [104] presented a model of grid field formation, where the grid emerges from the contrast between the space continuity and the neuronal fatigue firing. The grid is built as the result of Hebbian learning in the feed-forward connections of their model. Different oscillatory-interference models have been also proposed to encode the speed and direction of motion information into a grid pattern in the frequency of the membrane-potential oscillators [73, 177], or the connections between the oscillations and the stability of the grid representations [102, 192]. Studies performed afterward also showed that GCs are modulated by local directional signals [74].

5.2 Full navigation models

Some works in the literature are based on the computational models proposed for the specialized cells presented in Section 5.1, building structures that replicate the functioning of brain regions for navigation. For example, in [111] the authors proposed a method for localization based on PCs. The whole processing pipeline consists of a first stage that detects and assigns landmarks building an internal map of PCs that represents the scenario, and a second stage that represents the relationships between the distances and the PCs' firing, using a Radial Basis Function (RBF) neural network [84]. Another model that combines PCs and GCs, applied to goal navigation learning was presented by [191]. The interest of this work relies on how a simulated agent is capable of, after learning environmental cues during several executions, finding a path to the aim by optimizing it at every iteration, merging the PCs and GCs learning behavior. The learning rule of the model consists of a winner-takes-all mechanism, where the cell with the maximal firing rate at a particular position wins the learning chance [105]. Similarly, [35] presented another goal-oriented brain inspired navigation model. In this work, the robot is able to converge to an optimal path by the guidance of population activities and reward signals. A grid-like representation for flexible navigation in challenging environments combined with theoretical models of GCs and Vector-based Navigation algorithms was proposed in [8]. They also showed how their algorithm was able to exploit optimal routes when combined with a path-based barrier avoidance strategy.

Other approaches propose models that better integrate the different cells such as [81], with a model for PC-GC generation that could be used as the foundations for the construction of cognitive maps. In this work, PCs and GCs generated from environment visual information are called *visual-PCs* and *visual-GCs* respectively. Then, the fusion of the *visual-GCs* with self-motion cues is called *united-GCs*, while PCs generated via supervised learning are called *united-PCs*. The first stage of the processing pipeline is the simulation of *visual-PCs* using a Gaussian function on the absolute location information; new cells are created during the exploration process every time the firing of the last activated cell goes below a fixed threshold. The firing rate depends on the distance from the explored location to the boundary. In the second stage, these cells generate the *visual-GCs* through a feed-forward network [30] to build their hexagonal firing pattern distribution. Next, in the third stage, *visual-GCs* are combined with self-motion cues using a genetic algorithm [97], thus generating *united-GCs*. In previous steps, since cells only use the boundary information as input, GCs cannot expand to the whole space. However, when including the self-motion information, the firing field can now be extended with additional information,

not limited by the boundaries. Finally, in the fourth stage, *united-PCs* are built by combining *united-GCs* and *visual-PCs* to improve the position accuracy of PCs away from the scene boundaries. This is done via a supervised fuzzy ART network [29], which is a competitive classifying and clustering network that implements plasticity and enables *continual learning*, so it has the ability to learn new knowledge without suffering from *catastrophic forgetting*.

Another set of models combined PCs and GCs with HDCs, improving tasks such as collision avoidance while navigating. Such a naive bio-inspired model was presented by [68, 69]. In this work, PCs are modeled so that the environment is encoded in *contexts* that contain currently observed visual cues given in polar coordinates. The PC associated with the current position fires by comparing the similarity of the current context to previous ones. For each context, a set of HDCs is created for the virtual rotations of the current context at a fixed angle. GCs are also modeled in a similar way to PCs and HDCs, by the similarity of displaced context, but in a specific activated PC context. GCs and HDCs contexts are used in combination to estimate the orientation and position of the robot. The model was validated on basic functions like mapping, guidance, homing, and shortcut finding, although is carried out in very simple scenarios.

Finally, some approaches also proposed models of the Lateral Entorhinal Cortex (LEC) and the Medial Entorhinal Cortex (MEC), regions that produce inputs for the hippocampus. In [199], the authors proposed a hierarchical network mimicking the LEC and MEC cortical areas for robot position prediction. In the proposed neural architecture the system learns the transformation of egocentric sensorimotor inputs into allocentric spatial representation for navigation, using a deep learning approach. The feed-forward connections, which convey spatiotemporal information, are learned by stochastic gradient descent. More recently, the same authors presented SeMINet [200], with a model that builds the map integrating self-motion and visual features.

5.2.1 Continuous Attractor Neural Networks (CANNs)

A special category of approaches for navigation is based on CANNs [160, 189]. These networks are dynamic models of cells imitating the behavior of the navigation neurons already mentioned in Section 5.1 [101]. They are built on cells connected to nearby cells by excitatory (positive) or reinforcing connections that allow energy in the central cell to propagate. Also, there are inhibitory (negative) connections from each cell to every other single cell in the network. With this inhibition, if a cell is activated, for example, a cell representing a specific orientation, it inhibits the activity in all the other cells representing other directions in the network by a certain amount. CANNs are considered to be a canon for the GCs modeling. The excitation between cells is usually represented via a Gaussian distribution. Following the formulation by [123], in a 2D-CANN of dimensions (n_x, n_y) , the excitation weights are computed as shown in Eq. 8. In this expression, δ_x and δ_y are constants of the variance for the 2D spatial distribution, and the distance between units cells is encoded by $u = (x - i)(\text{mod } n_x)$ and $v = (y - j)(\text{mod } n_y)$.

$$\varepsilon_{u,v}^{gc} = \frac{1}{\delta_x \sqrt{2\pi}} e^{\frac{-u^2}{2\delta_x^2}} \cdot \frac{1}{\delta_y \sqrt{2\pi}} e^{\frac{-v^2}{2\delta_y^2}} \quad (8)$$

Thus, for each cell (x, y) , the total excitation from every cell (i, j) is expressed as shown in Eq. 9.

$$\Delta P_{\mathbf{x}}^{gc} = \sum_i^{n_x} \sum_j^{n_y} P_{i,j}^{gc} \cdot \varepsilon_{u,v}^{gc} \quad (9)$$

And next, local and global inhibition is obtained as in Eq. 10.

$$\Delta P_{\mathbf{x}}^{gc} = \sum_i^{n_x} \sum_j^{n_y} P_{i,j}^{gc} \cdot \psi_{u,v}^{gc} - \varphi \quad (10)$$

Where $\psi_{u,v}^{gc}$ is an inhibitory weight matrix to update local inhibition activity, and φ is a constant to perform global inhibition equally to all the cells' activity, which must not be negative. And finally, the total activity in the CANN is normalized as shown in Eq. 11.

$$P_{\mathbf{x}}^{gc'} = \frac{P_{\mathbf{x}}^{gc}}{\sum_i^{n_x} \sum_j^{n_y} P_{i,j}^{gc}} \quad (11)$$

Many models in the literature use CANNs and some of the most popular were conceived by mimicking the neural behaviour underlying the navigation process in the rodent brain. The open-source navigation

RatSLAM model [7] is inspired by the rodent’s hippocampus and estimates position and orientation using visual cues from a single low-resolution camera. GCs are simulated by what they call a *Pose Cell Network*, which is indeed a 3D-CANN that encodes the pose of the agent given by (x, y, yaw) . These cells refine the pose estimation using previous computations but in a smooth and continuous transitional manner. The input to this network comes from two separate sources: 1) translational and rotational velocity from a simple visual odometry algorithm; 2) responses from *Local View Cells*, related to the PCs. These cells learn, for every new scenario, the activity of the *Pose Cell Network* at that moment, and inject this energy again to the network when revisiting a known place, a feature that is crucial for robot relocalization. Finally, loop-closure is performed using an experience map [122]. As proof of the potential of RatSLAM, it was able to successfully map an entire neighborhood, or to navigate in an office environment for over two weeks [124, 126].

NeuroSLAM [197] is a 4-Dof SLAM method built on GC and HDC models, integrated with a vision system that provides external visual cues and self-motion cues. This new model is inspired by 2D RatSLAM but extends it to 3D mapping and localization. In [36], the authors also introduced a model of PCs and HDCs based on CANNs to encode visual and inertial inputs, and then continuously compute the current orientation and position.

Biological path integration was originally modeled using vector navigation [27, 51]. In [52], the author proposed a GC-based CANN navigation system, since GCs are thought to be part of the path integration processes and to be responsible for building the coordinate system of the environment. The authors used CANNs that were able to perform path integration by using the inputs from GCs. However, vector navigation is also interesting, pointing to the direction from the current location to a specific goal or a previously visited location stored in memory. This is usually referred to as the goal-direction signal, and it is crucial for a robot navigating the real world. An improvement in long-range navigation was presented later in [53], with models that allow robots to navigate up to 100 meters in the experiments. More recently, the same author extended this work by integrating PCs and BVCs for navigation and local obstacle avoidance [54]. In this work, the authors highlight that while PCs are linked to a topological representation of the environment, GCs are thought to support metric navigation. Therefore, GCs alone are not enough for navigation in cluttered environments, where obstacle avoidance is required. Even when integrating BVCs, the robot may get stuck and PCs will be required to undo some of the previous steps and attempt a new trajectory, using what authors called an experience-dependent topological graph.

Finally, an applied work [34], shows a successful system that uses GCs and HDCs to model a localization method for Unmanned Aerial Vehicles (UAVs). HDCs are built into a 1D-CANN to model the yaw rotational velocity of the UAV and this result is fed to a 3D-CANN built with GCs. This second 3D-CANN estimates the UAV translational velocity, which is provided to a last stage that decodes position from the firings of GCs by using a period-adic method. The method is computationally simple, and experiments show robust estimations under noisy conditions.

6 Discussion

Event cameras are rapidly gaining popularity as a promising neuromorphic technology. They offer several advantages over frame cameras, such as low power consumption, low latency, and high dynamic range. With advances in control policies, agile robots can make aggressive maneuvers, but their perception and processing pipelines are limited by latency. However, event cameras have opened up new possibilities to push these limits further. They can efficiently compress visual information by suppressing redundant data, thereby reducing processing pipeline latency. This intrinsic ability of event cameras makes it possible to close perception-action loops in real-time specifically when using event-wise processing that exploits the asynchronous sparse nature of the event streams. SNNs have the potential to become the best-suited networks for event-driven processing, as these neural architectures have shown to be capable not only of processing asynchronous data streams but also of processing more efficiently than conventional networks. Moreover, SNNs are able to adapt their connections to the input over time, without any explicit reprogramming, a cognitive process called plasticity.

Regarding the computational models of areas of the brain involved in navigation, they are also gaining popularity since biology has been shown to efficiently perform navigation. The key aspect underlying this energy-efficient processing comes from the environment representation that is built and how this cognitive representation is interpreted by specialized navigation cells to determine the location within it. This type of representation is believed to be responsible for the low-power consumption of the brain when perceiving, reasoning, and acting during the navigation process; thus, it is valuable research for mobile robotics. To this end, this paper discusses future directions of neuromorphic engineering:

Exploiting the temporal information of event cameras: It can be noticed that most current solutions try to readapt the traditional computer vision algorithms by using synthetic virtual frames as input, aggregating events. The survey shows that there are better approaches to handling event data because temporal information is partially or completely ignored. Some other approaches have tried to redesign this paradigm representation, proposing alternatives such as volumes of events, time surfaces, etc. However, there has yet to be an established event representation or processing technique. Indeed, the works that present better results are those that use a combination of different sensors. This means that there is still much work to be done with these novel cameras.

ANNs vs. SNNs: Regarding the state-of-the-art methodology for event cameras, on the one hand, for most applications, deep learning methods lead the way. For example, neural networks offer more potential to infer dense optical flow than traditional methods. This is because their architecture is based on multi-scale models, which can abstract global and local scene information using a 3D representation of events or even directly using the stream of events. On the other hand, model-based methods just started to emerge for ego-motion estimation. However, these architectures are purely von Neumann, which means they take advantage of grid-like representations given synchronously at a fixed time interval; new mechanisms that exploit the asynchronous nature of the streams of events are required. Our review shows that SNNs still lack accuracy compared to ANNs. However, this gap is being reduced and in fact, some task-specific SNNs already perform better. The reader should keep in mind that SNNs provide efficient information encoding, leading to reduced computational complexity and energy consumption. Finally, SNNs are the way of asynchronously processing spikes or events when implemented in neuromorphic hardware. However, there is still a long way to go before scalable processing architectures and hardware can be used to solve real-world problems.

Overcoming the limitations of CANNs: The current state-of-the-art computational models that replicate brain areas focus on studying specialized navigation cells modeled by CANNs. CANN-based models have been shown to successfully perform navigation in unstructured environments in experiments in real scenarios. The problem with the CANNs is that they need to be more scalable and are used to solve simple problems such as direction estimation. Moreover, regarding SLAM, a bio-inspired method for loop closure is still needed to be successful.

Compromise between energy and accuracy: In terms of energy efficiency, neuromorphic computing has been demonstrated to outperform deep learning approaches. However, they still lack accuracy performance and energy efficiency is only attainable when implemented in neuromorphic computing platforms. This combination could raise neuromorphic computing to a new stage as the successor of current deep learning approaches. Combining neuromorphic sensory, computing platforms, and neural algorithms would mean a new step towards effective biologically-inspired robotics with lower latency, higher energy efficiency, and lower inference times.

Brain-inspired 3D navigation methods: Many of the navigation approaches inspired by the brain spacial cells architecture tackle only the 2D SLAM, and most of them are very naive as the functional relationship between brain cells is still not entirely clear. We hope that recent discoveries of 3D spatial reasoning in the brain will lead to new and promising 3D brain-inspired SLAM, as presented by NeuroSLAM [197].

7 Conclusions

This survey has covered research on bio-inspired methods for robotics applications, focusing on vision-based navigation tasks. First, we have provided an overview of the global pipeline in autonomous robotic systems, from perception to actuation. Although this survey has presented several brain-inspired perception approaches and sensors, it remains a great challenge to develop efficient end-to-end systems that also target control and actuation following bio-inspired principles, with promising advances in fields such as active vision that pairs perception and control.

Bio-inspired techniques are gaining popularity in robotics due to long-standing evidence that animals operate more effectively than state-of-the-art navigation algorithms. We conclude that, despite recent bio-inspired breakthroughs, the paradigm of the perfect autonomous system is still a long way off from reality. But it is getting close, as neuromorphic sensors or brain-inspired computing platforms have shown us. Moreover, some navigation algorithms based on the brain cognitive representation and scene understanding, are coming out. Although naive, these solutions are the first steps toward models inspired by the human brain. Finally, in response to the initial question we presented in this paper “*Can neuromorphic engineering be the successor of current and classical navigation solutions in robotics?*”, we anticipate that, eventually, the advantages of brain-inspired navigation algorithms will combine with the

ones from the event-based sensors approaches, providing an innovative approach for effective bio-inspired autonomous robotics.

Acknowledgements This work was supported by the Spanish National Grant PID2022-141466OB-I00 funded by MICIU/AEI/10.13039/501100011033 and by ERDF/EU.

References

- [1] Himanshu Akolkar, SioHoi Ieng, and Ryad Benosman. Real-time high speed motion prediction using fast aperture-robust event-driven visual flow, 2018. URL <https://arxiv.org/abs/1811.11135>.
- [2] Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, et al. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE T. on Computer-aided design of integrated circuits and systems*, 34(10):1537–1557, 2015.
- [3] Iñigo Alonso and Ana C. Murillo. Ev-segnet: Semantic segmentation for event-based cameras. 2018. doi:10.48550/ARXIV.1811.12039. URL <https://arxiv.org/abs/1811.12039>.
- [4] Scientific American. The brain’s gps tells you where you are and where you’ve come from, 2016. URL <https://www.scientificamerican.com/article/the-brain-s-gps-tells-you-where-you-are-and-where-you-ve-come-from/>.
- [5] Stephen A. Baccus, Bence P. Ölveczky, Mihai Manu, and Markus Meister. A retinal circuit that computes object motion. *The Journal of Neuroscience*, 28:6807 – 6817, 2008. URL <https://api.semanticscholar.org/CorpusID:3470745>.
- [6] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004.
- [7] David Ball, Scott Heath, Janet Wiles, Gordon Wyeth, Peter Corke, and Michael Milford. Openrationslam: an open source brain-based slam system. *Autonomous Robots*, 34:1–28, 04 2013. doi:10.1007/s10514-012-9317-9.
- [8] Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin Chadwick, Thomas Degris, Joseph Modayil, Greg Wayne, Hubert Soyer, Fabio Viola, Brian Zhang, Ross Goroshin, Neil Rabinowitz, Razvan Pascanu, Charlie Beattie, Stig Petersen, and Dharshan Kumaran. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557, 05 2018. doi:10.1038/s41586-018-0102-6.
- [9] Patrick Bardow, Andrew J. Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 884–892, 2016.
- [10] Levick WR Barlow HB. The mechanism of directionally selective units in rabbit’s retina. *The Journal of physiology*, 178(3):477–504, 1965. doi:doi:10.1113/jphysiol.1965.sp007638.
- [11] Francisco Barranco, Cornelia Fermüller, and Yiannis Aloimonos. Contour motion estimation for asynchronous event-driven cameras. *Proceedings of the IEEE*, 102(10):1537–1556, 2014.
- [12] Francisco Barranco, Cornelia Fermüller, and Yiannis Aloimonos. Bio-inspired motion estimation with event-driven sensors. In *International Work-Conference on Artificial Neural Networks*, pages 309–321. Springer, 2015.
- [13] Francisco Barranco, Cornelia Fermüller, Yiannis Aloimonos, and Tobi Delbrück. A dataset for visual navigation with neuromorphic methods. *Frontiers in Neuroscience*, 10, 02 2016. doi:10.3389/fnins.2016.00049.
- [14] Andrew J. Barry, Peter R. Florence, and Russ Tedrake. High-speed autonomous obstacle avoidance with pushbroom stereo. *Journal of Field Robotics*, 35(1):52–68, 2018.

- [15] Caswell Barry, Colin Lever, R. Hayman, Tom Hartley, Stephen Burton, John O’Keefe, Kate Jeffery, and Neil Burgess. The boundary vector cell model of place cell firing and spatial memory. *Reviews in the neurosciences*, 17:71–97, 02 2006.
- [16] Ryad Benosman, Sio-Hoï Sio-Hoï Ieng, Paul Rogister, and Christoph Posch. Asynchronous event-based hebbian epipolar geometry. *IEEE Transactions on Neural Networks*, 22(11):1723–1734, 2011.
- [17] Ryad Benosman, Sio-Hoi Ieng, Charles Clercq, Chiara Bartolozzi, and Mandyam Srinivasan. Asynchronous frameless event-based optical flow. *Neural Networks*, 27:32–37, 2012. ISSN 0893-6080. doi:<https://doi.org/10.1016/j.neunet.2011.11.001>. URL <https://www.sciencedirect.com/science/article/pii/S0893608011002930>.
- [18] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi. Event-based visual flow. *IEEE Transactions on Neural Networks and Learning Systems*, 25(2):407–417, 2014. doi:[10.1109/TNNLS.2013.2273537](https://doi.org/10.1109/TNNLS.2013.2273537).
- [19] Edgar Bermudez-Contreras, Benjamin J. Clark, and Aaron Wilber. The neuroscience of spatial navigation and the relationship to artificial intelligence. *Frontiers in Computational Neuroscience*, 14, 2020. ISSN 1662-5188.
- [20] Jonathan Binas, Daniel Neil, Shih-Chii Liu, and Tobi Delbruck. Ddd17: End-to-end davis driving dataset, 2017. URL <https://arxiv.org/abs/1711.01458>.
- [21] Tom Birkonen, Henning Winterfeld, Simon Fichtner, Adrian Petraru, and Hermann Kohlstedt. A spiking and adapting tactile sensor for neuromorphic applications. *Scientific reports*, 10(1):1–11, 2020.
- [22] Fabian Blöchliger, Marius Fehr, Marcin Dymczyk, Thomas Schneider, and Roland Y. Siegwart. Topomap: Topological mapping and navigation based on visual slam maps. *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1–9, 2018.
- [23] R. Bohlin and L.E. Kavraki. Path planning using lazy prm. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 1, pages 521–528 vol.1, 2000.
- [24] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240×180 130 db 3 μ s latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014.
- [25] Vincent Brebion, Julien Moreau, and Franck Davoine. Real-time optical flow for vehicular perception with low-and high-resolution event cameras. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):15066–15078, 2021.
- [26] Levi Burner, Anton Mitrokhin, Cornelia Fermüller, and Yiannis Aloimonos. Evimo2: An event camera dataset for motion segmentation, optical flow, structure from motion, and visual inertial odometry in indoor scenes with monocular or stereo algorithms. *arXiv*, 05 2022.
- [27] Daniel Bush, Caswell Barry, Daniel Manson, and Neil Burgess. Using grid cells for navigation. *Neuron*, 87, 2015.
- [28] Luis Alejandro Camuñas-Mesa, Teresa Serrano-Gotarredona, and Bernabé Linares-Barranco. Event-driven sensing and processing for high-speed robotic vision. In *IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 516–519, 2014.
- [29] Gail A. Carpenter, Stephen Grossberg, and David B. Rosen. Fuzzy art: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4(6):759–771, 1991. ISSN 0893-6080.
- [30] Luísa Castro and Paulo Aguiar. A feedforward model for the formation of a grid field where spatial information is provided solely from place cells. *Biological cybernetics*, 108, 02 2014. doi:[10.1007/s00422-013-0581-3](https://doi.org/10.1007/s00422-013-0581-3).
- [31] Andrea Censi and Davide Scaramuzza. Low-latency event-based visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 703–710, 2014. doi:[10.1109/ICRA.2014.6906931](https://doi.org/10.1109/ICRA.2014.6906931).

- [32] William Chamorro, Joan Solà, and Juan Andrade-Cetto. Event-based line slam in real-time. *IEEE Robotics and Automation Letters*, 7(3):8146–8153, 2022. doi:10.1109/LRA.2022.3187266.
- [33] Chin-Wei Chang, Wei-Ting Chen, and Ming-Tzu Ho. Implementation of time-optimal motion planning for scara robots. In *2020 International Automatic Control Conference (CACS)*, pages 1–6, 2020. doi:10.1109/CACS50047.2020.9289744.
- [34] Lijun Chao, Zhi Xiong, Jianye Liu, Chuang Yang, and Yudi Chen. A brain-inspired localization system for the uav based on navigation cells. *Aircraft Engineering and Aerospace Technology*, 93: 1221–1228, 09 2021.
- [35] Qiuying Chen and Hongwei Mo. A brain-inspired goal-oriented robot navigation system. *Applied Sciences*, 9(22), 2019. ISSN 2076-3417. doi:10.3390/app9224869. URL <https://www.mdpi.com/2076-3417/9/22/4869>.
- [36] Yudi Chen, Zhi Xiong, Jianye Liu, Chuang Yang, Lijun Chao, and Yang Peng. A positioning method based on place cells and head-direction cells for inertial/visual brain-inspired navigation system. *Sensors*, 21:7988, 11 2021.
- [37] Elisabetta Chicca, Michael Schmuker, and Martin P Nawrot. Neuromorphic sensors, olfaction., 2014.
- [38] François Chollet. Xception: Deep learning with depthwise separable convolutions, 2016. URL <https://arxiv.org/abs/1610.02357>.
- [39] Fabien Colonnier, Luca Della Vedova, and Garrick Orchard. Espee: Event-based sensor pose estimation using an extended kalman filter. *Sensors*, 21(23), 2021. ISSN 1424-8220. doi:10.3390/s21237840. URL <https://www.mdpi.com/1424-8220/21/23/7840>.
- [40] Frontiers Science Communications. Infographic: How grid cells in the brain help us navigate the world, 2021. URL <https://blog.frontiersin.org/2021/11/05/grid-cells-brain-navigate-may-britt-moser/>.
- [41] N Correll, B Hayes, C Heckman, and A Roncone. Introduction to autonomous robots: Mechanisms. *Sensors, Actuators, and Algorithms*. MIT Press, Cambridge, MA,, 2022.
- [42] Michael Csorba, Jeffrey K Uhlmann, and Hugh F Durrant-Whyte. New approach to simultaneous localization and dynamic map building. In *Navigation and Control Technologies for Unmanned Systems*, volume 2738, pages 26–36. SPIE, 1996.
- [43] Manon Dampfhofer, Thomas Mesquida, Alexandre Valentian, and Lorena Anghel. Are snns really more energy-efficient than anns? an in-depth hardware-aware study. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(3):731–741, 2023. doi:10.1109/TETCI.2022.3214509.
- [44] Simon Davidson and Steve B Furber. Comparison of artificial and spiking neural networks on digital hardware. *Frontiers in Neuroscience*, 15:651141, 2021.
- [45] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1):82–99, 2018.
- [46] Daniel Deniz, Eduardo Ros, Cornelia Fermuller, and Francisco Barranco. When do neuromorphic sensors outperform cameras? learning from dynamic features. In *57th Annual Conference on Information Sciences and Systems (CISS)*, 2023.
- [47] Peter U Diehl and Matthew Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in computational neuroscience*, 9:99, 2015.
- [48] Peter U Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International joint conference on neural networks (IJCNN)*, pages 1–8. ieee, 2015.
- [49] Weiheng Liu Dongqing Zou, Feng Shi and Hyunsurk Eric Ryu. Robust dense depth maps generations from sparse dvs stereos. In *British Machine Vision Conference (BMVC)*, pages 39.1–39.11, 2017.

- [50] Marco Dorigo, Guy Theraulaz, and Vito Trianni. Swarm Robotics: Past, Present, and Future [Point of View]. *Proceedings of the IEEE*, 109(7):1152–1165, July 2021. doi:10.1109/JPROC.2021.3072740. URL <https://hal.science/hal-03362874>.
- [51] Vegard Edvardsen. A passive mechanism for goal-directed navigation using grid cells. volume ECAL 2015: the 13th European Conference on Artificial Life of ALIFE 2022: The 2022 Conference on Artificial Life, pages 191–198, 07 2015.
- [52] Vegard Edvardsen. Long-range navigation by path integration and decoding of grid cells in a neural network. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 4348–4355, 2017. doi:10.1109/IJCNN.2017.7966406.
- [53] Vegard Edvardsen. Goal-directed navigation based on path integration and decoding of grid cells in an artificial neural network. *Natural Computing: An International Journal*, 18(1):13–27, mar 2019. ISSN 1567-7818.
- [54] Vegard Edvardsen, Andrej Bicanski, and Neil Burgess. Navigating with grid and place cells in cluttered environments. *Hippocampus*, 30(3):220–232, 2020.
- [55] Davide Falanga, Suseong Kim, and Davide Scaramuzza. How fast is too fast? the role of perception latency in high-speed sense and avoid. *IEEE Robotics and Automation Letters*, 4(2):1884–1891, 2019.
- [56] Davide Falanga, Kevin Kleber, and Davide Scaramuzza. Dynamic obstacle avoidance for quadrotors with event cameras. *Science Robotics*, 5(40):eaaz9712, 2020.
- [57] Thomas Finateu, Atsumi Niwa, Daniel Matolin, Koya Tsuchimoto, Andrea Mascheroni, Etienne Reynaud, Pooria Mostafalu, Frederick T. Brady, Ludovic Chotard, Florian Le Goff, Hirotsugu Takahashi, Hayato Wakabayashi, Yusuke Oike, and Christoph Posch. 5.10 a 1280×720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86 μ m pixels, 1.066geps readout, programmable event-rate controller and compressive data-formatting pipeline. *2020 IEEE International Solid-State Circuits Conference - (ISSCC)*, pages 112–114, 2020.
- [58] Tobias Fischer and Michael Milford. How many events do you need? event-based visual place recognition using sparse but varying pixels. *IEEE Robotics and Automation Letters*, 7(4):12275–12282, 2022.
- [59] Philipp Foehn, Angel Romero, and Davide Scaramuzza. Time-optimal planning for quadrotor waypoint flight. *Science Robotics*, 6(56):eabh1221, 2021.
- [60] Christian Forster, Matia Pizzoli, and Davide Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15–22, 2014. doi:10.1109/ICRA.2014.6906584.
- [61] Justas Furmonas, John Liobe, and Vaidotas Barzdenas. Analytical review of event-based camera depth estimation methods and systems. *Sensors*, 22(3), 2022. ISSN 1424-8220. doi:10.3390/s22031201. URL <https://www.mdpi.com/1424-8220/22/3/1201>.
- [62] Guillermo Gallego, Jon Lund, Elias Mueggler, Henri Rebecq, Tobi Delbrück, and Davide Scaramuzza. Event-based, 6-dof camera tracking from photometric depth maps. *IEEE T. on Pat. Anal. and Mach. Intel.*, 40(10):2402–2412, 2018.
- [63] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3867–3876, 2018. doi:10.1109/CVPR.2018.00407.
- [64] Guillermo Gallego, Mathias Gehrig, and Davide Scaramuzza. Focus is all you need: Loss functions for event-based vision. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12272–12281, 2019.
- [65] Guillermo Gallego, Tobi Delbrück, G. Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J. Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. Event-based vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44:154–180, 2022.

- [66] Ling Gao, Yuxuan Liang, Jiaqi Yang, Shaoxun Wu, Chenyu Wang, Jiaben Chen, and Laurent Kneip. Vector: A versatile event-centric benchmark for multi-sensor slam. *IEEE Robotics and Automation Letters*, 7(3):8217–8224, 2022.
- [67] Philippe Gaussier, Jean Paul Banquet, Nicolas Cuperlier, Mathias Quoy, Lise Aubin, Pierre-Yves Jacob, Francesca Sargolini, Etienne Save, Jeffrey L. Krichmar, Bruno Poucet, Basil el Jundi, Almut Kelber, and Barbara Webb. Merging information in the entorhinal cortex: what can we learn from robotics experiments and modeling? *Journal of Experimental Biology*, 222, 02 2019. ISSN 0022-0949. doi:10.1242/jeb.186932.
- [68] Simon Gay, Kévin Le Run, Edwige Pissaloux, Katerine Romeo, and Christèle Lecomte. Towards a predictive bio-inspired navigation model. *Information*, 12(3), 2021. ISSN 2078-2489. doi:10.3390/info12030100. URL <https://www.mdpi.com/2078-2489/12/3/100>.
- [69] Simon Gay, Edwige Pissaloux, and Jean-Paul Jamont. *A Bio-Inspired Model for Robust Navigation Assistive Devices: A Proof of Concept*, pages 17–33. 03 2023. doi:10.1007/978-3-031-29548-5_2.
- [70] Daniel Gehrig, Michelle Rüegg, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. Combining events and frames using recurrent asynchronous multimodal networks for monocular depth prediction. *IEEE Robotics and Automation Letters*, 6(2):2822–2829, 2021. doi:10.1109/LRA.2021.3060707.
- [71] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 2021. doi:10.1109/LRA.2021.3068942.
- [72] Mathias Gehrig, Mario Millhäuser, Daniel Gehrig, and Davide Scaramuzza. E-raft: Dense optical flow from event cameras. In *2021 International Conference on 3D Vision (3DV)*, pages 197–206, 2021. doi:10.1109/3DV53792.2021.00030.
- [73] Caroline Geisler, David Robbe, Michaël Zugaro, Anton Sirota, and György Buzsáki. Hippocampal place cell assemblies are speed-controlled oscillators. *Proceedings of the National Academy of Sciences*, 104(19):8149–8154, 2007.
- [74] Klara Gerlei, Jessica Passlack, Ian Hawes, Brianna Vandrey, Holly Stevens, Ioannis Papastathopoulos, and Matthew F Nolan. Grid cells are modulated by local head direction. *Nature communications*, 11(1):1–14, 2020.
- [75] Suman Ghosh and Guillermo Gallego. Multi-event-camera depth estimation and outlier rejection by refocused events fusion. *Advanced Intelligent Systems*, page 2200221, 2022.
- [76] Roddy M. Grieves and Kate J. Jeffery. The representation of space in the brain. *Behavioural Processes*, 135:113–131, 2017. ISSN 0376-6357. doi:<https://doi.org/10.1016/j.beproc.2016.12.012>. URL <https://www.sciencedirect.com/science/article/pii/S0376635716302480>.
- [77] Antea Hadžiger, Igor Cvišić, Ivan Marković, Sacha Vražić, and Ivan Petrović. Feature-based event stereo visual odometry. In *2021 European Conference on Mobile Robots (ECMR)*, pages 1–6, 2021. doi:10.1109/ECMR50962.2021.9568811.
- [78] Germain Haessig, Andrew Cassidy, Rodrigo Alvarez, Ryad Benosman, and Garrick Orchard. Spiking optical flow for event-based sensors using ibm’s truenorth neurosynaptic system. *IEEE TBioCAS*, 12(4):860–870, 2018.
- [79] Germain Haessig, Xavier Berthelon, Sio-Hoi Ieng, and Ryad Benosman. A spiking neural network model of depth from defocus for event-based neuromorphic vision. *Scientific Reports*, 9:3744, 03 2019.
- [80] Jesse Hagenaars, Federico Paredes-Vallés, and Guido De Croon. Self-supervised learning of event-based optical flow with spiking neural networks. *Advances in Neural Information Processing Systems*, 34:7167–7179, 2021.

- [81] Kun Han, Dewei Wu, and Lei Lai. A brain-inspired adaptive space representation model based on grid cells and place cells. *Computational Intelligence and Neuroscience*, 2020:1–12, 08 2020. doi:10.1155/2020/1492429.
- [82] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
- [83] Tom Hartley, N. Burgess, C. Lever, F. Cacucci, and J. O’Keefe. Modeling place fields in terms of the cortical inputs to the hippocampus. *Hippocampus*, 10(4):369–379, 2000.
- [84] Alfa Heryudono, Elisabeth Larsson, Alison Ramage, and Lina Sydow. Preconditioning for radial basis function partition of unity methods. *J. Sci. Comput.*, 67(3):1089–1109, jun 2016. ISSN 0885-7474. doi:10.1007/s10915-015-0120-6. URL <https://doi.org/10.1007/s10915-015-0120-6>.
- [85] Javier Hidalgo-Carrió, Daniel Gehrig, and Davide Scaramuzza. Learning monocular dense depth from events. In *2020 International Conference on 3D Vision (3DV)*, pages 534–542. IEEE, 2020.
- [86] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1):185–203, 1981. ISSN 0004-3702.
- [87] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic mapping framework based on octrees. *Autonomous Robots*, 2013. <https://octomap.github.io>.
- [88] Junjie Hu, Yan Zhang, and Takayuki Okatani. Visualization of convolutional neural networks for monocular depth estimation. pages 3868–3877, 10 2019. doi:10.1109/ICCV.2019.00397.
- [89] Yuhuang Hu, Shih-Chii Liu, and Tobi Delbruck. v2e: From video frames to realistic dvs events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1312–1321, 2021.
- [90] Kate J. Jeffery. Spatial cognition: Entorhinal cortex and the hippocampal place-cell map. *Current Biology*, 25(24):R1181–R1183, 2015. ISSN 0960-9822. doi:<https://doi.org/10.1016/j.cub.2015.10.048>. URL <https://www.sciencedirect.com/science/article/pii/S0960982215013111>.
- [91] Hyun Ho Jeon and Yun-Ho Ko. Lidar data interpolation algorithm for visual odometry based on 3d-2d motion estimation. In *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–2, 2018.
- [92] Yuqian Jiang, Shiqi Zhang, Piyush Khandelwal, and Peter Stone. Task planning in robotics: an empirical comparison of pddl-based and asp-based systems. 2018. doi:10.48550/ARXIV.1804.08229.
- [93] Damien Joubert, Alexandre Marcireau, Nic Ralph, Andrew Jolley, André van Schaik, and Gregory Cohen. Event camera simulator improvements via characterized parameters. *Frontiers in Neuroscience*, 15, 2021. ISSN 1662-453X.
- [94] Eric R Kandel, James H Schwartz, Thomas M Jessell, Steven Siegelbaum, A James Hudspeth, Sarah Mack, et al. *Principles of neural science*, volume 4. McGraw-hill New York, 2000.
- [95] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *International Journal of Robotic Research - IJRR*, 30:846–894, 06 2011. doi:10.1177/0278364911406761.
- [96] Karthik Karur, Nitin Sharma, Chinmay Dharmatti, and Joshua E. Siegel. A survey of path planning algorithms for mobile robots. *Vehicles*, 3(3):448–468, 2021. ISSN 2624-8921. doi:10.3390/vehicles3030027. URL <https://www.mdpi.com/2624-8921/3/3/27>.
- [97] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. A review on genetic algorithm: Past, present, and future. *Multimedia Tools and Applications*, 80(5):8091–8126, feb 2021. ISSN 1380-7501. doi:10.1007/s11042-020-10139-6. URL <https://doi.org/10.1007/s11042-020-10139-6>.

- [98] L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, Aug 1996. ISSN 2374-958X.
- [99] Hanme Kim, Stefan Leutenegger, and Andrew J. Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. *Lecture Notes in Computer Science*, 9910 LNCS:349 – 364, 2016.
- [100] Simon Klenk, Jason Chui, Nikolaus Demmel, and Daniel Cremers. Tum-vie: The tum stereo visual-inertial event dataset. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8601–8608. IEEE, 2021.
- [101] James J. Knierim and Kechen Zhang. Attractor dynamics of spatially correlated neural activity in the limbic system. *Annual Review of Neuroscience*, 35(1):267–285, 2012.
- [102] Julie Koenig, Ashley N Linder, Jill K Leutgeb, and Stefan Leutgeb. The spatial periodicity of grid cells is not sustained during reduced theta oscillations. *Science*, 332(6029):592–595, 2011.
- [103] Adarsh Kosta and Kaushik Roy. Adaptive-spikenet: Event-based optical flow estimation using spiking neural networks with learnable neuronal dynamics. *arXiv*, 09 2022. doi:10.48550/arXiv.2209.11741.
- [104] Emilio Kropff and Alessandro Treves. The emergence of grid cells: Intelligent design or just adaptation? *Hippocampus*, 18(12):1256–1269, 2008. doi:<https://doi.org/10.1002/hipo.20520>. <https://onlinelibrary.wiley.com/doi/pdf/10.1002/hipo.20520>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/hipo.20520>.
- [105] Tomas Kulvicius, Minija Tamasiunaite, James Ainge, Paul Dudchenko, and Florentin Wörgötter. Odor supported place cell model and goal navigation in rodents. *Journal of computational neuroscience*, 25(3):481–500, 2008.
- [106] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram Shi, and Ryad Benosman. Hots: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE T on Patt. Analysis and Mach. Intell.*, 39(7):1346–1359, 2017.
- [107] Steven M. LaValle. Rapidly-exploring random trees : a new tool for path planning. *The annual research report*, 1998.
- [108] Chankyu Lee, Adarsh Kosta, Alex Zihao Zhu, Kenneth Chaney, Kostas Daniilidis, and Kaushik Roy. Spike-flownet: Event-based optical flow estimation with energy-efficient hybrid neural networks. In *ECCV*, page arXiv:2003.06696, 2020.
- [109] Chankyu Lee, Adarsh Kumar Kosta, and Kaushik Roy. Fusion-flownet: Energy-efficient optical flow estimation using sensor fusion and deep fused spiking-analog network architectures. pages 6504–6510, 2021.
- [110] Colin Lever, Stephen Burton, Ali Jeewajee, John O’Keefe, and Neil Burgess. Boundary vector cells in the subiculum of the hippocampal formation. *Journal of Neuroscience*, 29(31):9771–9777, 2009. ISSN 0270-6474.
- [111] Weilong Li, Dewei Wu, Yang Zhou, and Jia Du. A bio-inspired method of autonomous positioning using spatial association based on place cells firing. *International Journal of Advanced Robotic Systems*, 14:172988141772801, 09 2017.
- [112] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbrück. A 128×128 120 db 15 μ s latency asynchronous temporal contrast vision sensor. *Solid-State Circuits, IEEE Journal of*, 43:566 – 576, 03 2008.
- [113] Xiuhong Lin, Chenhui Yang, Xuesheng Bian, Weiquan Liu, and Cheng Wang. Eagan: Event-based attention generative adversarial networks for optical flow and depth estimation. *IET Computer Vision*, pages n/a–n/a, 06 2022.
- [114] Shih-Chii Liu, Tobi Delbrück, Giacomo Indiveri, Adrian Whatley, and Rodney Douglas. *Event-based neuromorphic systems*. John Wiley & Sons, 2014.

- [115] Siyu Liu and Bo Ma. Stereo visual odometry with information enhancement at feature points. In *2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*, pages 654–661, 2021.
- [116] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021. URL <https://arxiv.org/abs/2103.14030>.
- [117] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Int. Joint Conference on Artificial Intelligence, IJCAI'81*, page 674–679, 1981.
- [118] Ren C. Luo and Wei Shih. Autonomous mobile robot intrinsic navigation based on visual topological map. In *2018 IEEE 27th International Symposium on Industrial Electronics (ISIE)*, pages 541–546, 2018. doi:10.1109/ISIE.2018.8433588.
- [119] Camille Mazzara, Albert Comelli, and Michele Migliore. Place cell’s computational model. In *Image Analysis and Processing. ICIAP 2022 Workshops*, pages 398–407, 2022.
- [120] Etienne Meunier, Anaïs Badoual, and Patrick Bouthemy. Em-driven unsupervised learning for efficient motion segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–12, 2022.
- [121] Chenglin Miao, Qichen Cao, Hiroshi T. Ito, Homare Yamahachi, Menno P. Witter, May-Britt Moser, and Edvard I. Moser. Hippocampal remapping after partial inactivation of the medial entorhinal cortex. *Neuron*, 88(3):590–603, 2015. ISSN 0896-6273.
- [122] Michael Milford and Ashley George. *Featureless Visual Processing for SLAM in Changing Outdoor Environments*, volume 92, pages 569–583. 01 2014. doi:10.1007/978-3-642-40686-7_38.
- [123] Michael Milford and Gordon Wyeth. Hippocampal models for simultaneous localisation and mapping on an autonomous robot. 2003.
- [124] Michael Milford and Gordon Wyeth. Persistent navigation and mapping using a biologically inspired slam system. *The International Journal of Robotics Research*, 29, 08 2010. doi:10.1177/0278364909340592.
- [125] Michael Milford, Hanme Kim, Stefan Leutenegger, and Andrew J. Davison. Towards visual slam with event-based cameras. 2015.
- [126] Michael J. Milford and Gordon F. Wyeth. Mapping a suburb with a single camera using a biologically inspired slam system. *IEEE Transactions on Robotics*, 24(5):1038–1053, 2008. doi:10.1109/TR.2008.2004520.
- [127] Michael J. Milford and Gordon. F. Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *2012 IEEE International Conference on Robotics and Automation*, pages 1643–1649, 2012.
- [128] Anton Mitrokhin, Cornelia Fermüller, Chethan Parameshwara, and Yiannis Aloimonos. Event-based moving object detection and tracking. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, oct 2018.
- [129] Anton Mitrokhin, Chengxi Ye, Cornelia Fermüller, Yiannis Aloimonos, and Tobi Delbrück. Evimo: Motion segmentation dataset and learning pipeline for event cameras. In *Conf. on Intelligent Robots and Systems (IROS)*, pages 6105–6112, 2019.
- [130] Anton Mitrokhin, Zhiyuan Hua, Cornelia Fermüller, and Yiannis Aloimonos. Learning visual motion segmentation using event surfaces. pages 14402–14411, 06 2020. doi:10.1109/CVPR42600.2020.01442.
- [131] Edvard I Moser and May-Britt Moser. A metric for space. *Hippocampus*, 18(12):1142–1156, 2008.
- [132] Edvard I Moser, May-Britt Moser, and Bruce L McNaughton. Spatial representation in the hippocampal formation: a history. *Nature neuroscience*, 20(11):1448–1464, 2017.

- [133] May-Britt Moser, David C Rowland, and Edvard I Moser. Place cells, grid cells, and memory. *Cold Spring Harbor perspectives in biology*, 7(2):a021808, 2015.
- [134] Elias Mueggler, Basil Huber, and Davide Scaramuzza. Event-based, 6-dof pose tracking for high-speed maneuvers. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2761–2768, 2014.
- [135] Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing*, 36(6):51–63, 2019.
- [136] Anh Nguyen, Thanh-Toan Do, and Nikos Caldwell, Darwinand Tsagarakis. Real-time 6dof pose relocalization for event cameras with stacked spatial lstm networks. In *CVPRW*, pages 1638–1645, 2019.
- [137] Xiaoqi Nong and Simon Hadfield. Asl-slam: An asynchronous formulation of lines for slam with event sensors. In *2022 Int. Conf. on Industrial Engineering and Applications*, page 84–91, 2022.
- [138] A. Novo, Liang Lu, and Pascual Campoy. Fast rrt* 3d-sliced planner for autonomous exploration using mavs. *Unmanned Systems*, 10(02):175–186, 2022.
- [139] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. Voxblox: Incremental 3d euclidean signed distance fields for on-board MAV planning. In *2017 Int. Conf. on Intelligent Robots and Systems (IROS)*, sep 2017.
- [140] Chethan M. Parameshwara, Simin Li, Cornelia Fermüller, Nitin J. Sanket, Matthew S. Evanusa, and Yiannis Aloimonos. Spikems: Deep spiking neural network for motion segmentation, 2021. URL <https://arxiv.org/abs/2105.06562>.
- [141] Chethan M Parameshwara, Nitin J Sanket, Chahat Deep Singh, Cornelia Fermüller, and Yiannis Aloimonos. 0-mms: Zero-shot multi-motion segmentation with a monocular event camera. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9594–9600. IEEE, 2021.
- [142] Federico Paredes-Vallés and Guido CHE de Croon. Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy. In *Conf. on Computer Vision and Pattern Recognition*, pages 3446–3455, 2021.
- [143] Scott Drew Pendleton, Hans Andersen, Xinxin Du, Xiaotong Shen, Malika Meghjani, You Hong Eng, Daniela Rus, and Marcelo H. Ang. Perception, planning, control, and coordination for autonomous vehicles. *Machines*, 5(1), 2017. ISSN 2075-1702.
- [144] E Piatkowska, A N Belbachir, and M Gelautz. Cooperative and asynchronous stereo vision for dynamic vision sensors. *Measurement Science and Technology*, 25(5):055108, apr 2014. doi:10.1088/0957-0233/25/5/055108. URL <https://dx.doi.org/10.1088/0957-0233/25/5/055108>.
- [145] Albert Pumarola, Alexander Vakhitov, Antonio Agudo, Alberto Sanfeliu, and Francesc Moreno-Noguer. Pl-slam: Real-time monocular visual slam with points and lines. In *2017 Int. Conf. on Robotics and Automation (ICRA)*, pages 4503–4508, 2017.
- [146] Ulysse Rançon, Javier Cuadrado-Anibarro, Benoit R. Cottreau, and Timothée Masquelier. Stereospike: Depth learning with a spiking neural network. *IEEE Access*, 10:127428–127439, 2022. doi:10.1109/ACCESS.2022.3226484.
- [147] Henri Rebecq, Timo Horstschaefner, Guillermo Gallego, and Davide Scaramuzza. Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Rob. and Automation Letters*, 2(2):593–600, 2017.
- [148] Henri Rebecq, Timo Horstschaefner, and Davide Scaramuzza. Real-time visual-inertial odometry for event cameras using keyframe-based nonlinear optimization. In *British Machine Vision Conference*, pages 1–8, September 2017.
- [149] Henri Rebecq, Guillermo Gallego, Elias Mueggler, and Davide Scaramuzza. Emvs: Event-based multi-view stereo-3d reconstruction with an event camera in real-time. *Int. J. Comput. Vision*, 126(12):1394–1414, dec 2018. ISSN 0920-5691.

- [150] Henri Rebecq, Daniel Gehrig, and Davide Scaramuzza. Esim: an open event camera simulator. In *Conference on robot learning*, pages 969–982. PMLR, 2018.
- [151] Nicoletta Risi, Enrico Calabrese, and Giacomo Indiveri. Instantaneous stereo depth estimation of real-world stimuli with a neuromorphic stereo-vision setup. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2021.
- [152] Robin Ritz, Mark W. Müller, Markus Hehn, and Raffaello D’Andrea. Cooperative quadrocopter ball throwing and catching. In *2012 IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pages 4972–4978, 2012.
- [153] Paul Rogister, Ryad Benosman, Sio-Hoi Ieng, Patrick Lichtsteiner, and Tobi Delbruck. Asynchronous event-based binocular stereo matching. *IEEE Transactions on Neural Networks and Learning Systems*, 23(2):347–353, 2012.
- [154] Angel Romero, Robert Penicka, and Davide Scaramuzza. Time-optimal online replanning for agile quadrotor flight. *IEEE Robotics and Automation Letters*, 7(3):7730–7737, 2022. doi:10.1109/LRA.2022.3185772.
- [155] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. volume 9351, pages 234–241, 10 2015. doi:10.1007/978-3-319-24574-4_28.
- [156] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High Speed Scenarios. September 2017. 1709.06310.
- [157] Edward Rosten, Reid B. Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:105–119, 2010.
- [158] Bodo Rueckauer and Tobi Delbruck. Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor. *Frontiers in Neuroscience*, 10, 2016. ISSN 1662-453X. doi:10.3389/fnins.2016.00176. URL <https://www.frontiersin.org/articles/10.3389/fnins.2016.00176>.
- [159] Talabattula Sai Abhishek, Daniel Schilberg, and Arockia Arockia Doss. Obstacle avoidance algorithms: A review. *IOP Conference Series: Materials Science and Engineering*, 1012:012052, 01 2021.
- [160] Alexei Samsonovich and Bruce L. McNaughton. Path integration and cognitive mapping in a continuous attractor neural network model. *Journal of Neuroscience*, 17(15):5900–5920, 1997. ISSN 0270-6474.
- [161] Nitin J. Sanket, Chethan M. Parameshwara, Chahat Deep Singh, Ashwin V. Kuruttukulam, Cornelia Fermüller, Davide Scaramuzza, and Yiannis Aloimonos. Evdodgenet: Deep dynamic obstacle dodging with event cameras. 2019.
- [162] Cedric Scheerlinck, Henri Rebecq, Daniel Gehrig, Nick Barnes, Robert E. Mahony, and Davide Scaramuzza. Fast image reconstruction with an event camera. In *IEEE Winter Conf. on App. of Computer Vision (WACV)*, pages 156–163, 2020.
- [163] Catherine Schuman, Shruti Kulkarni, Maryam Parsa, J. Mitchell, Prasanna Date, and Bill Kay. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, 2:10–19, 01 2022.
- [164] Shikai Shao, Chenglong He, Yuanjie Zhao, and Xiaojing Wu. Efficient trajectory planning for uavs using hierarchical optimization. *IEEE Access*, 9:60668–60681, 2021. doi:10.1109/ACCESS.2021.3073420.
- [165] Shintaro Shiba, Yoshimitsu Aoki, and Guillermo Gallego. Secrets of event-based optical flow. 2022.
- [166] Nir Shlezinger, Jay Whang, Yonina C Eldar, and Alexandros G Dimakis. Model-based deep learning. *Proceedings of the IEEE*, 2023.

- [167] Mennatullah Siam, Sepehr Valipour, Martin Jagersand, and Nilanjan Ray. Convolutional gated recurrent networks for video segmentation. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3090–3094, 2017.
- [168] Shay Snyder, Hunter Thompson, Md Abdullah-Al Kaiser, Gregory Schwartz, Akhilesh Jaiswal, and Maryam Parsa. Object motion sensitivity: A bio-inspired solution to the ego-motion problem for event-based cameras, 2023.
- [169] Lea Steffen, Daniel Reichard, Jakob Weinland, Jacques Kaiser, Arne Roennau, and Rüdiger Dillmann. Neuromorphic stereo vision: A survey of bio-inspired sensors and algorithms. *Frontiers in Neurorobotics*, 13, 2019. ISSN 1662-5218.
- [170] Yunjae Suh, Seungnam Choi, Masamichi Ito, Jeongseok Kim, Youngho Lee, Jong-Mo Seo, Heejae Jung, Dong-Hee Yeo, Seol Namgung, Jong-Su Bong, Sehoon Yoo, Seung-Hun Shin, Doowon Kwon, Pilkyu Kang, Seokho Kim, Hoonjoo Na, Kihyun Hwang, Chang-Woo Shin, Jun-Seok Kim, Paul K. J. Park, Joonseok Kim, Hyunsuk Eric Ryu, and Yongin Park. A 1280×960 dynamic vision sensor with a $4.95\text{-}\mu\text{m}$ pixel pitch and motion artifact minimization. *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5, 2020. URL <https://api.semanticscholar.org/CorpusID:224958352>.
- [171] Haixin Sun, Minh-Quan Dao, and Vincent Fremont. 3d-flownet: Event-based optical flow estimation with 3d representation. In *2022 IEEE Intelligent Vehicles Symposium (IV)*, pages 1845–1850, 2022. doi:10.1109/IV51971.2022.9827380.
- [172] JS Taube, RU Muller, and JB Ranck. Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. *Journal of Neuroscience*, 10(2):420–435, 1990. ISSN 0270-6474.
- [173] Mohammad-Hassan Tayarani-Najaran and Michael Schmuker. Event-based sensing and signal processing in the visual, auditory, and olfactory domain: a review. *Frontiers in Neural Circuits*, 15:610446, 2021.
- [174] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow (extended abstract). *ArXiv*, pages 4839–4843, 08 2021. doi:10.24963/ijcai.2021/662.
- [175] Justin Thomas, Giuseppe Loianno, Joseph Polin, Koushil Sreenath, and Vijay Kumar. Toward autonomous avian-inspired grasping for micro aerial vehicles*. *Bioinspiration & Biomimetics*, 9(2):025010, may 2014.
- [176] Jesus Tordesillas and Jonathan P. How. PANTHER: Perception-aware trajectory planner in dynamic environments. *IEEE Access*, 10:22662–22677, 2022. doi:10.1109/access.2022.3154037. URL <https://doi.org/10.1109%2Faccess.2022.3154037>.
- [177] Roger D Traub, Miles A Whittington, and Mark O Cunningham. Simulation of oscillatory dynamics induced by an approximation of grid cell output. *Reviews in the Neurosciences*, 2022.
- [178] Konstantinos Tsintotas, Loukas Bampis, and Antonios Gasteratos. The revisiting problem in simultaneous localization and mapping: A survey on visual loop closure detection. *IEEE T. on Intelligent Transportation Systems*, 23(11):1–12, 2022.
- [179] Anup Vanarse, Adam Osseiran, and Alexander Rassau. Neuromorphic engineering—a paradigm shift for future im technologies. *IEEE Instrumentation & Measurement Magazine*, 22(2):4–9, 2019.
- [180] Anup Vanarse, Adam Osseiran, Alexander Rassau, and Peter van der Made. A hardware-deployable neuromorphic solution for encoding and classification of electronic nose data. *Sensors*, 19(22):4831, 2019.
- [181] Valentina Vasco, Arren Glover, and Chiara Bartolozzi. Fast event-based harris corner detection exploiting the advantages of event-driven cameras. pages 4144–4149, 10 2016. doi:10.1109/IROS.2016.7759610.
- [182] Ziwei Wang, Yonhon Ng, Cedric Scheerlinck, and Robert Mahony. An asynchronous kalman filter for hybrid event cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 448–457, 2021.

- [183] David Weikersdorfer and Jörg Conradt. Event-based particle filtering for robot self-localization. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 866–870, 2012. doi:[10.1109/ROBIO.2012.6491077](https://doi.org/10.1109/ROBIO.2012.6491077).
- [184] David Weikersdorfer, Raoul Hoffmann, and Jorg Conradt. Simultaneous localization and mapping for event-based vision systems. volume 7963, pages 133–142, 09 2013. doi:[10.1109/978-1-4673-642-3_14](https://doi.org/10.1109/978-1-4673-642-3_14).
- [185] David Weikersdorfer, David B. Adrian, Daniel Cremers, and Jörg Conradt. Event-based 3d slam with a depth-augmented dynamic vision sensor. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 359–364, 2014.
- [186] P.J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.
- [187] James CR Whittington, David McCaffary, Jacob JW Bakermans, and Timothy EJ Behrens. How to build a cognitive map. *Nature Neuroscience*, pages 1–16, 2022.
- [188] Rey Reza Wiyatno, Anqi Xu, and Liam Paull. Lifelong topological visual navigation. *IEEE Robotics and Automation Letters*, 7(4):9271–9278, 2022. doi:[10.1109/LRA.2022.3189164](https://doi.org/10.1109/LRA.2022.3189164).
- [189] S Wu, KYM Wong, CCA Fung, Y Mi, and W Zhang. Continuous attractor neural networks: Candidate of a canonical model for neural information representation. *F1000Research*, 5(156), 2016.
- [190] L. Xu and E. Oja. Randomized hough transform (rht): Basic mechanisms, algorithms, and computational complexities. *CVGIP: Image Understanding*, 57(2):131–154, 1993. ISSN 1049-9660. doi:<https://doi.org/10.1006/cium.1993.1009>. URL <https://www.sciencedirect.com/science/article/pii/S1049966083710090>.
- [191] Chuankui Yan, Rubin Wang, Jingyi Qu, and Guanrong Chen. Locating and navigation mechanism based on place-cell and grid-cell models. *Cognitive Neurodynamics*, 10:353–360, 2016.
- [192] Yijia Yan, Neil Burgess, and Andrej Bicanski. A model of head direction and landmark coding in complex environments. *PLoS computational biology*, 17(9):e1009434, 2021.
- [193] Jawad N. Yasin, Sherif A. S. Mohamed, Mohammad-hashem Haghbayan, Jukka Heikkonen, Hannu Tenhunen, Muhammad Mehboob Yasin, and Juha Plosila. Night vision obstacle detection and avoidance based on bio-inspired vision sensors. In *2020 IEEE SENSORS*, pages 1–4, 2020. doi:[10.1109/SENSORS47125.2020.9278914](https://doi.org/10.1109/SENSORS47125.2020.9278914).
- [194] Chengxi Ye, Anton Mitrokhin, Cornelia Fermüller, James A Yorke, and Yiannis Aloimonos. Unsupervised learning of dense optical flow, depth and egomotion with event-based sensors. In *IROS*, pages 5831–5838. IEEE, 2020.
- [195] Zihan Yin, Md Abdullah-Al Kaiser, Lamine Ousmane Camara, Mark Camarena, Maryam Parsa, Ajey Jacob, Gregory Schwartz, and Akhilesh Jaiswal. Iris: Integrated retinal functionality in image sensors. *Frontiers in Neuroscience*, 17, 2023. ISSN 1662-453X.
- [196] Khalid Yousif, Alireza Bab-Hadiashar, and Reza Hoseinnezhad. An overview to visual odometry and visual slam: Applications to mobile robotics. *Intelligent Industrial Systems*, 1, 11 2015. doi:[10.1007/s40903-015-0032-7](https://doi.org/10.1007/s40903-015-0032-7).
- [197] Fangwen Yu, Jiang Shang, Youjian Hu, and Michael Milford. Neuroslam: a brain-inspired slam system for 3d environments. *Biological Cybernetics*, 113:515 – 545, 2019.
- [198] Jiqing Zhang, Bo Dong, Haiwei Zhang, Jianchuan Ding, Felix Heide, Baocai Yin, and Xin Yang. Spiking transformers for event-based single object tracking. In *Conf. on Comp. Vision and Pattern Recognition (CVPR)*, pages 8801–8810, June 2022.
- [199] Dongye Zhao, Bailu Si, and Xiaoli Li. Learning allocentric representations of space for navigation. *Neurocomputing*, 453:579–589, 2021. ISSN 0925-2312. doi:<https://doi.org/10.1016/j.neucom.2020.10.013>. URL <https://www.sciencedirect.com/science/article/pii/S0925231220315083>.

- [200] Dongye Zhao, Zheng Zhang, Hong Lu, Sen Cheng, Bailu Si, and Xisheng Feng. Learning cognitive map representations for navigation by sensory–motor integration. *IEEE Transactions on Cybernetics*, 52(1):508–521, 2022.
- [201] Junwei Zhao, Shiliang Zhang, Lei Ma, Zhaofei Yu, and Tiejun Huang. Spikingsim: A bio-inspired spiking simulator. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 3003–3007, 2022.
- [202] Yi Zhou, Guillermo Gallego, Henri Rebecq, Laurent Kneip, Hongdong Li, and Davide Scaramuzza. Semi-dense 3d reconstruction with a stereo event camera. In *Proceedings of the European conference on computer vision (ECCV)*, pages 235–251, 2018.
- [203] Yi Zhou, Guillermo Gallego, and Shaojie Shen. Event-based stereo visual odometry. *IEEE Transactions on Robotics*, 37(5):1433–1450, 2021. doi:[10.1109/TRO.2021.3062252](https://doi.org/10.1109/TRO.2021.3062252).
- [204] Alex Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE Rob. and Autom. Letters*, pages 2032–2039, 2018.
- [205] Alex Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Conf. on Comp. Vision and Pattern Recognition (CVPR)*, page arXiv:1812.08156, June 2018.
- [206] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based visual inertial odometry. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5816–5824, 2017. doi:[10.1109/CVPR.2017.616](https://doi.org/10.1109/CVPR.2017.616).
- [207] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *ArXiv*, abs/1802.06898, 2018.
- [208] Yi-Fan Zuo, Jiaqi Yang, Jiaben Chen, Xia Wang, Yifu Wang, and Laurent Kneip. Devo: Depth-event camera visual odometry in challenging conditions. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2179–2185, 2022.