

Review

Not peer-reviewed version

Application of Event Cameras and Neuromorphic Computing to VSLAM: A Survey

[Sangay Tenzin](#) , [Alexander Rassau](#) ^{*} , [Douglas Chai](#)

Posted Date: 16 May 2024

doi: 10.20944/preprints202405.1094.v1

Keywords: Simultaneous localization and mapping (SLAM); event camera; neuromorphic processing; robotics; autonomous systems; sensor fusion; real-time processing; machine vision



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Review

Application of Event Cameras and Neuromorphic Computing to VSLAM: A Survey

Sangay Tenzin, Alexander Rassau * and Douglas Chai

School of Engineering, Edith Cowan University, Western Australia; s.tenzin@ecu.edu.au;

a.rassau@ecu.edu.au; d.chai@ecu.edu.au

* Correspondence: a.rassau@ecu.edu.au

Abstract: Simultaneous Localization and Mapping (SLAM) is a crucial function for most autonomous systems, allowing them to both navigate through and create maps of unfamiliar surroundings. Traditional Visual SLAM, also commonly known as VSLAM, relies on frame-based cameras and structured processing pipelines, which face challenges in dynamic or low-light environments. However, recent advancements in event camera technology and neuromorphic processing offer promising opportunities to overcome these limitations. Event cameras inspired by biological vision systems capture the scenes asynchronously consuming minimal power but with higher temporal resolution. Neuromorphic processors, which are designed to mimic the parallel processing capabilities of the human brain, offer efficient computation for real-time data processing of event-based data streams. This paper provides a comprehensive overview of recent research efforts in integrating event cameras and neuromorphic processors into VSLAM systems. It discusses the principles behind event cameras and neuromorphic processors, highlighting their advantages over traditional sensing and processing methods. Furthermore, an in-depth survey was conducted on state-of-the-art approaches in event-based SLAM, including feature extraction, motion estimation, and map reconstruction techniques. Additionally, the integration of event cameras with neuromorphic processors, focusing on their synergistic benefits in terms of energy efficiency, robustness, and real-time performance was explored. The paper also discusses the challenges and open research questions in this emerging field, such as sensor calibration, data fusion, and algorithmic development. Finally, the potential applications and future directions for event-based SLAM systems are outlined, ranging from robotics and autonomous vehicles to augmented reality.

Keywords: Simultaneous localization and mapping (SLAM); event camera; neuromorphic processing; robotics; autonomous systems; sensor fusion; real-time processing; machine vision

1. Introduction

A wide and growing variety of robots are increasingly being employed in different indoor and outdoor applications. To support this, autonomous navigation systems have become essential for carrying out many of the required duties [1]. However, such systems must be capable of completing assigned tasks successfully and accurately with minimal human intervention. To increase the effectiveness and efficiency of such systems, they should be capable of navigating to a given destination while simultaneously updating their real-time location and developing a map of the surroundings. Towards this, Simultaneous Localization and Mapping (SLAM) is currently one of the most employed methods for localization and navigation of mobile robots [2]. The concept of SLAM has originated from the robotics and computer vision field. SLAM is a joint problem of simultaneously locating the position of the robots while developing a map of their surroundings [3]. It has become a critical technology to tackle the difficulties of allowing machines (autonomous systems) to independently navigate and map unfamiliar surroundings [3,4]. With SLAM, the location and map information of the autonomous systems will be continuously updated in real time. This process can help users in getting the status of the system as well as serve as a reference in making

autonomous navigation-related decisions [3]. It helps robots gain autonomy and reduce the requirement for human operation or intervention [3,4]. Moreover, with effective SLAM methods, mobile robots such as vacuum cleaners, autonomous vehicles, aerial drones, and others [2,4] can effectively navigate a dynamic environment autonomously.

The sensor choice affects the performance and efficacy of the SLAM solution [3] and should be decided based on the sensor's information-gathering capability, power cost and precision. The primary sensor types commonly utilized in SLAM applications are laser sensors (such as Light Detection and Ranging (LiDAR) sensors) and vision sensors. Laser-based SLAM typically offers higher precision; however, these systems tend to be more expensive and power-hungry [5]. Moreover, they lack semantic information and face challenges in loop closure detection. In environments with a lack of scene diversity, such as uniform corridors or consistently structured tunnels, degradation issues may arise, particularly affecting laser SLAM performance compared to Visual SLAM (VSLAM) [5]. Conversely, VSLAM boasts advantages in terms of cost-effectiveness, compact size, minimal power consumption, and the ability to perceive rich information, rendering it more suitable for indoor settings [5].

In recent decades, VSLAM has gained significant development attention as research has demonstrated that detailed scene information can be gathered from visual data [3,6] as well as due to the increased availability of low-cost cameras [6,7]. In VSLAM, cameras such as monocular, stereo and RGB-D are used to gather the information that is used for solving the localization and map-building problems. It records a continuous video stream by capturing frames of the surrounding environment at a specific rate. Generally, the classical VSLAM framework follows the steps as shown in Figure 1: sensor data acquisition, visual odometry (VO; also known as front-end), backend filtering/optimization, loop closure and reconstruction [8]. Sensor data acquisition involves the acquisition and pre-processing of data captured by the sensors (a camera in the case of VSLAM). VO is used for measuring the movement of the camera between the adjacent frames (ego-motion) and generating a rough map of the surroundings. The backend optimizes the camera pose received from VO and the result of loop closure in order to generate an efficient trajectory and map for the system. Loop closure determines if the system has previously visited the location to minimize the accumulated drift and update the backend for further optimization. With reconstruction, a map of the system will be developed based on the camera trajectory estimation.

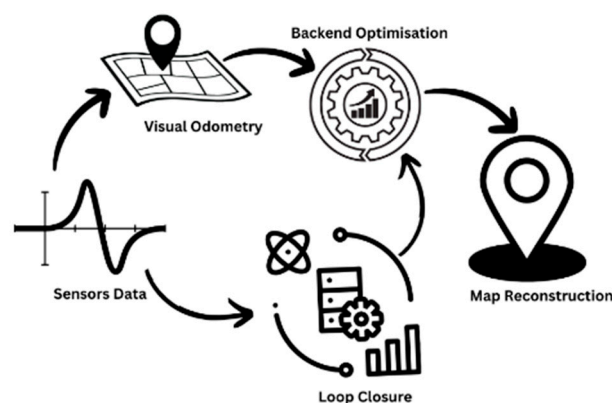


Figure 1. Classic visual SLAM framework.

Conventional VSLAM systems (based on traditional cameras) gather image data of fixed frames, which results in repetitive and often redundant information leading to high computational requirements and other drawbacks [9,10]. Further, they often fail to achieve the expected performance in challenging environments such as those with high dynamic ranges or light-changing conditions [9–15] due to constraints such as susceptibility to motion blur, high power consumption and low dynamic range. Following those limitations, research in emerging technologies of event cameras has evolved to attempt to address the issues of traditional VSLAM. The advent of novel concepts and the

production of bio-inspired visual sensors and processors through developments in neuroscience and neuromorphic technologies have brought a radical change in the processes of artificial visual systems [9,16,17]. An event camera (also known as a Dynamic Vision Sensor (DVS) or neuromorphic camera) operates very differently from conventional frame-based cameras; it only generates an output (in the form of timestamped events or spikes) when there are changes in the brightness of a scene [9,12,16,18,19]. Compared to regular cameras, event cameras have a greater dynamic range, reduced latency, higher temporal resolution, and significantly lower power consumption and bandwidth usage [3,9,12,13,17,18,20–23]. However, sensors based on these principles are relatively new to the market and their integration poses some challenges as new algorithms are needed as existing approaches are not directly applicable.

Similarly, in an attempt to further reduce the power cost, the research trends of mimicking the biological intelligence of the human brain and its behavior known as neuromorphic computing [11,24] are gaining more research focus to be applied in autonomous systems and robots as an extension to the use of event-based cameras for SLAM [24]. In neuromorphic computing, computational systems are designed by mimicking the composition and operation of the human brain. The objective is to create algorithms and hardware replicating the brain's energy efficiency and parallel processing capabilities [25]. Unlike von Neumann Computers, Neuromorphic Computers (also known as non-von Neumann computers) consist of neurons and synapses rather than a separate central processing unit (CPU) and memory units [26]. Moreover, as they are fully event-driven and highly parallel, in contrast to traditional computing systems, they can natively deal with spikes-based outputs rather than binary data [26]. Furthermore, the advent of neuromorphic processors with various sets of signals to mimic the behavior of biological neurons and synapses [11,27,28] has paved a new direction in the neuroscience field. This enables the hardware to asynchronously communicate between its components and the memory in an efficient manner, which results in less consumption of power in addition to other advantages [11,26,28]. As the computation is based on neural networks, it has become a primarily relevant platform for use in artificial intelligence and machine learning applications to enhance robustness and performance [26,29].

The combination of event cameras and neuromorphic processing, which takes inspiration from the efficiency of the human brain has the potential to offer a revolutionary approach to improving SLAM capabilities [9]. The use of event cameras allows SLAM systems to better handle dynamic situations and fast motion without being affected by motion blur or illumination variations. Event cameras provide high dynamic range imagery and low latency through asynchronous pixel-level brightness change capture [9,12,16,18,19]. Additionally, neuromorphic processors emulate the brain's structure and functionality [25], enabling efficient and highly parallel processing, which is particularly advantageous for real-time SLAM operations on embedded devices. This integration would facilitate improved perception, adaptability, and efficiency in SLAM applications, overcoming the limitations of conventional approaches and paving the way for more robust and versatile robotic systems [3]. The successful implementation of these trending technologies is expected to make smart and creative systems capable of making logical analyzes at the edge, further enhancing the productivity of the processes, improving precision and minimizing the exposure of humans to hazards [11,24,30].

Various reviews have been conducted on applying event cameras or neuromorphic computing approaches in robot vision and SLAM, but no review has been conducted on the integration of both technologies into a neuromorphic SLAM system. The reviews by [9,31] have primarily discussed event cameras and given only a brief introduction to both SLAM and the application of neuromorphic computing. Similarly, [11,28,32,33] have covered neuromorphic computing technology and its challenges, however, no clear direction towards its integration into event-based SLAM was provided. Review papers [19,22,34–38] have mentioned the methods and models to be employed in SLAM but did not discuss the combined approach.

The purpose of this paper is to provide a comprehensive review of potential VSLAM approaches based on event cameras and neuromorphic processors. With a thorough review of existing studies in VSLAM based on traditional cameras, the critical analysis of the methodologies and technologies

employed is discussed. Moreover, an overview of challenges and further directions to improve or resolve the mentioned problems of traditional VSLAM by the existing literature are presented. The review also covers the rationale and analysis of employing event cameras in VSLAM to address the issues faced by conventional VSLAM approaches. Furthermore, the feasibility of integrating neuromorphic processing into event-based SLAM to further enhance performance and power efficiency is explored.

The paper is organized as follows. Section 2 gives an overview of VSLAM and discusses the standard frame-based cameras and their limitations while employed in SLAM. In Section 3, event cameras and their working principle are presented including the general potential benefits and applications in various fields. Section 4 focuses on the application of neuromorphic computing in VSLAM that has the potential to address the performance and power issues faced by the autonomous system. Finally, a summary of key findings and identified future directions for VSLAM based on event cameras and neuromorphic processors can be found in Section 5.

2. Camera-Based SLAM (VSLAM)

For SLAM implementations, VSLAM is more popular than LiDAR-based SLAM for smaller-scale autonomous systems, particularly unmanned aerial vehicles (UAV), as it is compact, cost-effective and less power-intensive [3,6,8,22,35,39]. Unlike the laser-based systems, VSLAM employs various cameras such as monocular, stereo, and RGB-D cameras for capturing the surrounding scene and is being explored by researchers for implementation in autonomous systems and other applications [3,8,22,35,39]. It has gained popularity in the last decade as it has succeeded in retrieving detailed information (color, texture, and appearance) using low-cost cameras and some progress towards practical implementation in real environments has been made [3,6,9,22]. One prevalent issue encountered in VSLAM systems is the issue of cumulative drift [5]. Minor inaccuracies are produced with every calculation and optimization made by the front end of the SLAM system. These small errors accumulate over the extended durations of uninterrupted camera movement, which eventually causes the estimated trajectory to deviate from the real motion trajectory.

These traditional camera-based VSLAM systems have generally failed to achieve the expected performance in challenging environments such as those with high dynamic ranges or changing lighting conditions [9–15] due to constraints such as susceptibility to motion blur, noise susceptibility and low dynamic range among others. Moreover, the information gathered with traditional VSLAM is inadequate to fulfil the task of autonomous navigation, obstacle avoidance as well as the interaction needs of intelligent autonomous systems in dealing with the human environment [10].

In line with the growing popularity of VSLAM in the last decade, researchers have worked on designing improved algorithms towards making practical and robust solutions for SLAM a reality. However, most of the successfully developed algorithms such as MonoSLAM [40], PTAM [41], DTAM [42] and SLAM++ [43] have been developed for stationary environments, where it is assumed that the camera is the sole moving item in a static environment. This means they are not suitable for applications where both the scene and the object are dynamic [44], like autonomous vehicles and UAVs.

SLAM algorithms must depend on data generated by the sensors. Table 1 provides a summary of the range of VSLAM algorithms that have been developed for testing and implementation in SLAM systems and the different sensor modalities used by each.

Table 1. Classification of VSLAM algorithms/methods.

Year	Name	Sensors	Descriptions (Key Points)	Strength (Achievements)
2024	TextSLAM [45]	RGB-D	Text objects in the environment are used to extract semantic features	More accurate and robust even under challenging conditions
2023	HFNet-SLAM [7]	Monocular	Extension of ORB-SLAM3 (incorporates CNNs)	Performs better than ORB-SLAM3 (higher accuracy)
2022	SO-SLAM [46]	Monocular	Introduced object spatial constraints (object level map)	Proposed two new methods for object SLAM
2022	SDF-SLAM [47]	Monocular	Semantic deep fusion model with deep learning	Less absolute error than the state-of-the-art SLAM framework
2022	UV-SLAM [48]	Monocular	Vanishing points (line features) are used for structural mapping	Localization accuracy and mapping quality have improved
2021	RS-SLAM [49]	RGB-D	Employed semantic segmentation model	Both static and dynamic objects are detected
2021	RDMO-SLAM [50]	RGB-D	Semantic label prediction using dense optical flow	Reduce the influence of dynamic objects in tracking
2021	RDS-SLAM [51]	RGB-D	Extends ORB-SLAM3; Added semantic thread and a semantic-based optimization thread	Tracking thread is not required to wait for semantic information as novel threads run in parallel
2021	ORB-SLAM3 [52]	Monocular, Stereo and RGB-D	Perform visual, visual-inertial and multimap SLAM	Effectively exploits the data associations and boosts the system accuracy level
2020	Structure-SLAM [53]	Monocular	Decoupled rotation and translation estimation	Outperforms the state of the art on common SLAM benchmarks
2020	VPS-SLAM [54]	RGB-D	Combined low-level VO/VIO with planar surfaces	Provides better results than the state-of-the-art VO/VIO algorithms
2020	DDL-SLAM [44]	RGB-D	Dynamic object segmentation and background painting added to ORB-SLAM2	Dynamic objects detected utilizing semantic segmentation and multi-view geometry
2019	PL-SLAM [55]	Stereo	Combines point and line segments	The first open-source SLAM system with points and line segment features
2017	ORB-SLAM2 [56]	Monocular, Stereo and RGB-D	Complete SLAM system including map reuse, loop closing, and re-localization capabilities	Achieved state-of-the-art accuracy while evaluating 29 popular public sequences
2015	ORB-SLAM [57]	Monocular	Feature-based monocular SLAM system	Robust to motion clutter, allows wide baseline loop closing and re-localization

2014	LSD-SLAM [58]	Monocular	Direct monocular SLAM system	Achieved post-estimation accuracy and 3D environment reconstructions
2011	DTAM [42]	Monocular	Camera tracking and reconstruction based on a dense feature	Achieved real-time performance using the commodity GPU hardware
2007	PTAM [41]	Monocular	Estimate camera pose in an unknown scene	Accuracy and robustness have surpassed the state-of-the-art system
2007	MonoSLAM [40]	Monocular	Real-Time Single Camera SLAM	Recovered the 3D trajectory of a monocular camera

2.1. Types of VSLAM

For conventional VSLAM, cameras such as monocular, stereo, and RGB-D are considered a source of data that can be used by the algorithm [6]. Depending on the type of camera employed by the VSLAM system, they are commonly categorized as described below.

2.1.1. Monocular Camera SLAM

Monocular SLAMs are SLAM systems that rely on a single camera. Researchers have shown a great deal of interest in this kind of sensor configuration because it is thought to be easy to use and reasonably priced [8]. A 2D image of a 3D scene is captured by the camera during a monocular photo shoot. During this 2D projection process, the depth or measurement of the distance between the object and the camera is lost [8]. It is necessary to capture multiple frames while varying the camera's view angle to recover the 3D structure from the 2D projection of a monocular camera. Monocular SLAM works on a similar principle, utilizing the movement of the camera to estimate its motion as well as the sizes and distances of objects in the scene, which together make up the scene's structure [8].

The movement of the objects in the image as the camera moves creates pixel disparity. The relative distance of objects from the camera can be quantitatively determined by calculating this disparity. These measurements are not absolute, though; for example, one can discern whether objects in a scene are bigger or smaller than one another while viewing a movie, but one cannot ascertain the actual sizes of the objects [8]. A discrepancy between the actual trajectory and map and those derived from monocular SLAM estimation is caused by an unknown factor called scale ambiguity. This disparity results from the inability of monocular SLAM to ascertain the true scale purely from 2D images [8]. When monocular SLAM is used in practical applications, this restriction may lead to serious problems [8]. The inability of 2D images from a single camera to provide enough information to reliably determine depth is the main cause of this issue. Stereo or RGB-D cameras are utilized to overcome this restriction and produce real-scaled depth.

2.1.2. Stereo Camera SLAM

Two synchronized monocular cameras are placed apart by a predetermined distance, called the baseline, to form a stereo camera [8]. Based on the baseline distance, each pixel's 3D position can be ascertained, akin to the method employed in human vision. Stereo cameras are suitable for use in both indoor and outdoor environments because they can estimate depth without the need for additional sensing equipment by comparing images taken by the left and right cameras [6,8]. However, configuring and calibrating stereo cameras or multi-camera systems is a complex process. The camera resolution and baseline length also limit the accuracy and depth range of these systems [8]. Furthermore, to produce real-time depth maps, stereo matching and disparity calculation demand a large amount of processing power, which frequently calls for the use of GPU or FPGA accelerators [8]. Therefore, in many cutting-edge stereo camera algorithms, the computational cost continues to be a significant obstacle. The additional depth information that can be extracted from stereo camera setups can provide valuable additional information for VSLAM algorithms, overcoming the scale problem described earlier, but at a cost of significantly increased processing overhead.

2.1.3. RGB-D Camera SLAM

Since 2010, a camera type known as RGB-D cameras; also called depth cameras has become available. Like laser scanners, these cameras use Time-of-Flight (ToF) or infrared structured light to actively emit light onto an object and detect the reflected light to determine the item's distance from the camera [6,8]. Unlike stereo cameras, this method uses physical sensors rather than software, which drastically lowers the amount of computational power required [8]. Nevertheless, a lot of RGB-D cameras still have several drawbacks, like a short field of view, noisy data, a limited measuring range, interference from sunshine, and the inability to calculate distances to transparent objects [6,8].

RGB-D cameras are not the best option for outdoor applications and are mostly used for SLAM in indoor environments [6,8].

2.2. Limitations of Frame-Based Cameras in VSLAM

While there have been some significant successes in the development of VSLAM algorithms for traditional frame-based cameras, some limitations still exist, as described below.

- **Ambiguity in feature matching:** In feature-based SLAM, feature matching is considered a critical step. However, frame-based cameras face difficulty in capturing scenes with ambiguous features (e.g. plain walls). Moreover, data without depth information (as obtained from standard monocular cameras) makes it even harder for the feature-matching process to distinguish between similar features, which can lead to potential errors in data association.
- **Sensitivity to lighting conditions:** The sensitivity of traditional cameras to changes in lighting conditions affects the features and makes it more challenging to match features across frames consistently [6]. This can result in errors during the localization and mapping process.
- **Limited field of view:** The use of frame-based cameras can be limited due to their inherently limited field of view. This limitation becomes more apparent in environments with complex structures or large open spaces. In such cases, having multiple cameras or additional sensor modalities may become necessary to achieve comprehensive scene coverage, but this can lead to greatly increased computational costs as well as other complexities.
- **Challenge in handling dynamic environments:** Frame-based cameras face difficulties when it comes to capturing dynamic environments, especially where there is movement of objects or people. It can be challenging to track features consistently in the presence of moving entities, and other sensor types such as depth sensors or Inertial Measurement units (IMUs) must be integrated, or additional strategies must be implemented to mitigate those challenges. Additionally, in situations where objects in a scene are moving rapidly, particularly if the camera itself is on a fast-moving platform (e.g. a drone), then motion blur can significantly degrade the quality of captured frames unless highly specialized cameras are used.
- **High computational requirements:** Although frame-based cameras are typically less computationally demanding than depth sensors such as LiDAR, feature extraction and matching processes can still necessitate considerable computational resources, particularly for real-time applications.

3. Event Camera-Based SLAM

Event cameras have gained attention in the field of SLAM due to their unique properties, such as high temporal resolution, low latency, and high dynamic range. However, tackling the SLAM issue using event cameras has proven challenging due to the inapplicability or unavailability of traditional frame-based camera methods and concepts such as feature detection, matching, and iterative image alignment. Events, being fundamentally distinct from images (Figure 2 shows the differing output of frame-based cameras relative to event cameras), necessitate the development of novel SLAM techniques. The task has been to devise approaches that harness the unique advantages of event cameras demonstrating their efficacy in addressing challenging scenarios that are problematic for current frame-based cameras. A primary aim when designing the methods has been to preserve the low latency nature of event data thereby estimating a state for every new event. However, individual events lack sufficient data to create a complete state estimate, such as determining the precise position of a calibrated camera with six degrees of freedom (DoF). Consequently, the objective has shifted to enabling each event to independently update the system's state asynchronously [9].

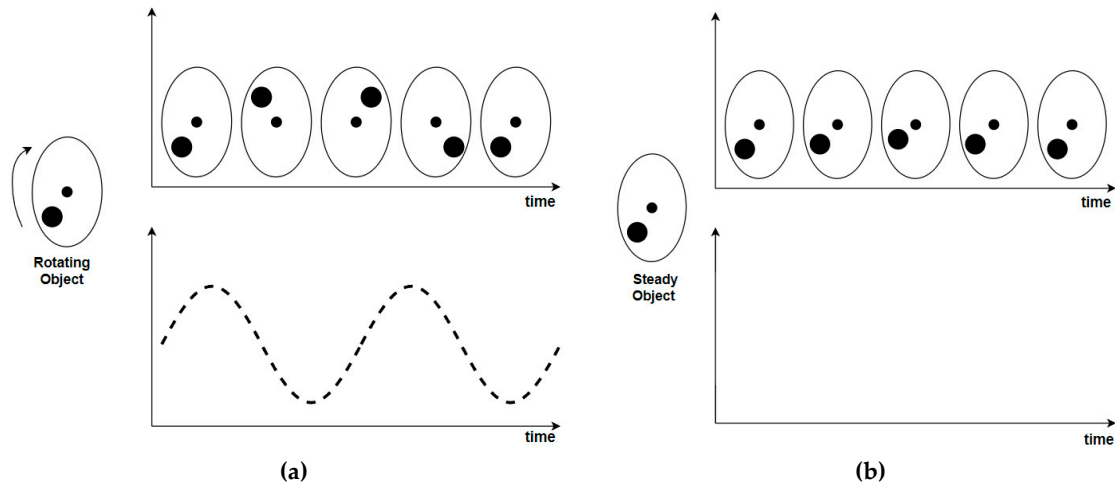


Figure 2. The output of frame-based camera (upper) vs event camera (lower): (a) With rotating object; (b) With the steady object (same brightness).

Utilizing event cameras' asynchrony and high temporal resolution, SLAM algorithms can benefit from reduced motion blur and improved visual perception in dynamic environments, ultimately leading to more robust and accurate mapping and localization results [19]. It can enhance the reconstruction of 3D scenes and enable tracking of fast motion with high precision. Furthermore, a low data rate and reduced power consumption compared to traditional cameras make them ideal for resource-constrained devices in applications such as autonomous vehicles, robotics, and augmented reality [19]. Moreover, it can be used to significantly increase the frame rate of low-framerate video while occupying significantly less memory space than conventional camera frames, enabling efficient and superior-quality video frame interpolation [22].

The integration of event cameras in SLAM systems opens new possibilities for efficient and accurate mapping, localization, and perception in dynamic environments, while also reducing power consumption and memory usage. Further, the utilization of event cameras in SLAM introduces exciting opportunities in terms of enhanced mapping and localization in dynamic environments.

3.1. Event Camera Operating Principles

Standard cameras and event cameras have significant differences when it comes to their working principle and operation [9,19,20]. Conventional cameras record a sequence of images at a predetermined frames per second (fps) rate, capturing intensity values for every pixel in every frame. On the other hand, event cameras record continuous-time event data, timestamped with microsecond resolution, with an event representing a detected change in pixel brightness [9,17,19]. Each pixel continuously updates the log intensity, and this is monitored for any notable changes in its value. If the value changes (either high or low) more than a certain threshold, an event will be generated [9].

An event is represented as a tuple, $e_k = (x_k, y_k, t_k, p_k)$, where (x_k, y_k) denotes the pixel coordinates that caused the event, t_k is the timestamp, and $p_k = \pm 1$ denotes the polarity or direction of the change in brightness [19]. Events are transferred from the pixel array to the peripheral and back again over a shared digital output bus, typically using the address-event representation (AER) readout technique [59]. Saturation of this bus, however, can occur and cause hiccups in the event transmission schedule. Event cameras' readout rates range from 2 MHz to 1,200 MHz, depending on the chip and hardware interface type being used [9].

Event cameras are essentially sensors that react to motion and brightness variations in a scene. They produce more events per second when there is greater motion. The reason is that every pixel modifies the rate at which it samples data using a delta modulator in response to variations in the log intensity signal it tracks. These sensors can respond to visual stimuli rapidly because of the sub-millisecond latency and microsecond precision timestamped events. Surface reflectance and scene lighting both affect how much light a pixel receives. A change in log intensity denotes a proportional

change in reflectance in situations where illumination is largely constant. The primary cause of these reflectance changes is the motion of objects in the field of view. Consequently, the brightness change events captured inherently possess an invariance to changes in scene illumination.

3.1.1. Event Generation Model

At each pixel position u_k the event camera sensor first records and stores the logarithmic intensity of brightness, or $L(u_k) = \log(I(u_k))$ and then continuously monitors this intensity value. The camera sensor at the pixel position $u_k = (x_k, y_k)$ generates an event, denoted by e_k , at time t_k when the difference in intensity, $\Delta L(u_k, t_k)$ exceeds a threshold, C , which is referred to as contrast sensitivity.

$$\Delta L(u_k, t_k) = L(u_k, t_k) - L(u_k, t_k - \Delta t_k) = p_k C \quad (1)$$

The last timestamp that was recorded in this context is $t_k - \Delta t_k$, which occurs when an event is triggered at the pixel u_k . The camera sensor then creates new events by iterating through the procedure to detect any changes in brightness at this pixel, updating the stored intensity value $L(u_k, t_k)$. The adjustable parameter C , or temporal contrast sensitivity, is essential to the camera's functioning. A high contrast sensitivity can result in fewer events produced by the camera and potential information loss, whereas a low contrast sensitivity, usually set between 10% and 15%, may cause an excessive number of noisy events [19].

3.1.2. Event Representation

The event camera records brightness variations at every pixel, producing a constant stream of event data. The low information content in each record and the sparse temporal nature of the data make the processing difficult. Filter-based methods directly process raw event data by combining it with sequential data, but they come with a high computational cost because they must update the camera motion for every new event. To mitigate this problem, alternative methods employ representations that combine event sequences and approximate camera motion for a collection of occurrences, achieving an equilibrium between computing cost and latency.

Common event representations for event-based VSLAM algorithms are described below:

Individual event: On an event-by-event basis, each event $e_k = (x_k, y_k, t_k, p_k)$ may be directly utilized in filter-based models, such as probabilistic filters [60] and Spiking Neural Networks (SNNs) [61]. With every incoming event, these models asynchronously change their internal states, either by recycling states from earlier events or by obtaining new information from outside sources, such as inertial data [9]. Although filter-based techniques can produce very little delay, they generally require a significant amount of processing power.

Packet: The event packet, also known as the point set, is an alternate representation used in event cameras. It stores an event data sequence directly in a temporal window of size N and is stated as follows:

$$\varepsilon = \{e_k\}_k^N = 1 \quad (2)$$

Event packets maintain specific details like polarity and timestamps, just like individual events do. Event packets facilitate batch operations in filter-based approaches [62] and streamline the search for the best answers in optimization methods [63,64] because they aggregate event data inside temporal frames. There are several variations of event packets, including event queues [63] and local point sets [64].

Event frame: A condensed 2D representation of an event that gathers data at a single pixel point is called an event frame. Assuming consistent pixel coordinates, this representation is achieved by transforming a series of events into an image-like format that is used as input for conventional frame-based SLAM algorithms [9].

Time surface: The Time Surface (TS), also called the Surface of Active Events (SAE), is a 2D representation in which every pixel contains a single time value, often the most recent timestamp of the event that occurred at that pixel [9]. A spatially structured visualization of the temporal data

related to occurrences throughout the camera's sensor array is offered by the time surface. Due to its ability to trace the time of events at different points on the image sensor, this representation can be helpful in a variety of applications, such as visual perception and motion analysis [9].

Motion-compensated event frame: A Motion-compensated event frame refers to a representation in event cameras where the captured events are aggregated or accumulated while compensating for the motion of the camera or objects in the scene [9]. Unlike traditional event frames that accumulate events at fixed pixel positions, motion-compensated event frames consider the dynamic changes in the scene over time. The events contributing to the frame are not simply accumulated at fixed pixel positions, but rather the accumulation is adjusted based on the perceived motion in the scene. This compensation can be performed using various techniques, such as incorporating information from inertial sensors, estimating camera motion, or using other motion models [9].

Voxel grid: A voxel grid can be used as a representation of 3D spatial information extracted from the events captured by the camera. Instead of traditional 2D pixel-based representations, a voxel grid provides a volumetric representation of the environment [9], allowing for 3D scene reconstruction, mapping, and navigation.

3D point set: Events within a spatiotemporal neighborhood are regarded as points in 3D space, denoted as $(x_k, y_k, t_k) \in \mathbb{R}$. Consequently, the temporal dimension is transformed into a geometric one. Plane fitting [65] or Point Net [66] are two point-based geometric processing methods that use this sparsely populated form.

Point sets on image plane: On the picture plane, events are viewed as a dynamic collection of 2D points. This representation is frequently used in early shape-tracking methods that use methods like mean-shift or iterative closest point (ICP) [67–71], in which the only information needed to follow edge patterns is events.

3.2. Method

To process the event, a relevant and valid method is required depending on the event representation and availability of the hardware platform. Moreover, the relevant information from event data can be extracted to fulfil the required task depending on the application and algorithm being utilized [9]. However, the efficacy of such efforts varies significantly based on the nature of the application and the unique demands it places on the data being extracted [9]. Figure 4 presents the overview of common methods used for event-based SLAM.

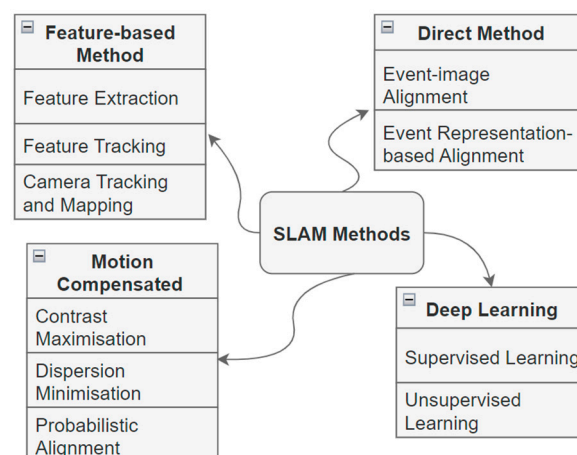


Figure 4. Common methods used in event-based SLAM.

3.2.1. Feature-Based Methods

The feature-based VSLAM algorithms comprise two main elements: (1) the extraction and tracking of features, and (2) the tracking and mapping of the camera. During the feature extraction

phase, resilient features, immune to diverse factors such as motion, noise, and changes in illumination, are identified. The ensuing feature tracking phase is employed to link features that correspond to identical points in the scene. Leveraging these associated features, algorithms for camera tracking and mapping concurrently estimate the relative poses of the camera and the 3D landmarks of the features.

3.1.1.1. Feature Extraction

Methods for feature extraction identify shape primitives within the event stream, encompassing features based on points and lines. Point-based features denote points of significance, such as the intersection of event edges. Various methods for point-based feature extraction, particularly corners, in the context of event cameras, have been used in the last decade or so. Traditional techniques involve employing algorithms like local plane fitting [72,73], frame-based corner detectors (e.g., eHarris [74], CHEC [75], eFAST [76]), and extensions of the Harris and FAST detectors to different event representations [77–79]. These methods, however, suffer from computational complexity, sensitivity to motion changes, and susceptibility to noise in event cameras [19]. To address these challenges, learning-based approaches [80,81] have been proposed, including the use of speed-invariant time surfaces and recurrent neural networks (RNNs) to enhance corner detection stability by implicitly modelling motion-variant patterns and event noise.

On the other hand, line-based features consist of clusters of events situated along straight lines. Several algorithms including classical methods like the Hough transformation and Line Segment Detector (LSD) [82] have been employed. Some approaches leverage spatiotemporal relationships [83] in event data, while others use an external IMU [84] data to group events. Examples include a spiking Hough transformation algorithm using spiking neurons [85] and extending the Hough transformation to a 3D point-based map [86] for improved performance. Event-Based VO with Point and Line features (PL-EVIO); leverages line-based event features to add more structure and constraint while efficiently handling the point-based event and picture characteristics). [87] directly applies the LSD algorithm to motion-compensated event streams, while the Event-based Line Segment Detector (ELiSeD) [88] computes event orientation using the Sobel filter. Other methods use optical flow [89] or plane-fitting algorithms [83] to cluster events and extract lines, demonstrating different techniques for line-based feature extraction from event data.

3.1.1.2. Feature Tracking

When working with event-based data, feature-tracking algorithms are utilized to link events to the relevant features. Motion trajectories, locations, and 2D rigid body transformations are examples of parametric models of feature templates that these algorithms update [19]. Methods include parametric transformations like the Euclidean transformation and descriptor matching for feature correspondences. Deep learning approaches use neural networks to predict feature displacements. Euclidean transformations model positions and orientations of event-based features, and tracking involves ICP algorithms [90] with enhancements like Euclidean distance weighting and 2D local histograms to improve accuracy and reduce drift. Some trackers, such as the Feature Tracking using Events and Frames (EKLT) tracker [91], align local patches of the brightness incremental image from event data with feature patterns and estimate brightness changes using the linearized Edge Gradient Method (EGM). Feature tracking often involves modelling feature motions on the image plane, with methods using expectation-maximization (EM) optimization steps [77,78] and the Lucas-Kanade (LK) optical flow tracker [79,87]. Continuous curve representations, like Bezier curves [92] and B-splines [93], are explored to address linear model assumptions. Multi-hypothesis methods [63,94] are proposed to handle event noise by discretizing spatial neighborhoods into hypotheses based on distance and orientation. Various techniques include using feature descriptors for direct correspondence establishment and building graphs with nodes representing event characteristics for tracking based on their discrete positions on the image plane [95,96]. Traditional linear noise models are contrasted with deep learning methods that implicitly model event noise [97].

3.1.1.3. Camera Tracking and Mapping

VSLAM algorithms, particularly those adapted for event-based tracking and mapping introduce two main paradigms: one where 3D maps are initialized, and tracking and mapping are performed in parallel threads and another where tracking and mapping are carried out simultaneously through joint optimization. The former offers computational efficiency, while the latter helps prevent drift errors. Event-based VSLAM approaches in camera tracking and mapping are categorized into four types: conventional frame-based methods, filter-based methods, continuous-time camera trajectory methods, and spatiotemporal consistency methods.

Conventional frame-based methods adapt existing VSLAM algorithms for event-based tracking and mapping using 2D image-like event representation. Various techniques, such as reprojection error and depth estimation, are employed for camera pose estimation. Event-based Visual Inertial Odometry (EVIO) [78] methods utilize IMU pre-integration and sliding-window optimization. Filter-based methods handle asynchronous event data using a state defined as the current camera pose and a random diffusion model as the motion model. These methods correct the state using error measurements, with examples incorporating planar features and event occurrence probabilities. Line-based SLAM methods update filter states during camera tracking and use the Hough transformation to extract 3D lines. Continuous-time camera trajectory methods represent the camera trajectory as a continuous curve, addressing the parameterization challenge faced by filter-based methods. Joint optimization methods based on incremental Structure from Motion (SfM) are proposed to update control states and 3D landmarks simultaneously. Spatiotemporal consistency methods introduce a constraint for events under rotational camera motion, optimizing motion parameters through iterative searches and enforcing spatial consistency using the trimmed ICP algorithm.

3.2.2. Direct Method

Direct methods do not require explicit data association, as opposed to feature-based approaches, and instead directly align event data in camera tracking and mapping algorithms. Although frame-based direct approaches use pixel intensities between selected pixels in source and target images to estimate relative camera poses and 3D positions, they are not applicable to event streams because of their asynchronous nature and the absence of brightness change information in the event data. Two kinds of event-based direct techniques: event-image alignment and event representation-based alignment have been developed to overcome this difficulty. The Edge Gradient Method (EGM) is used by event-image alignment techniques, such as those demonstrated by [60,98], to take advantage of the photometric link between brightness variations from events and absolute brightness in images. Event representation-based alignment techniques [15,99] use spatiotemporal information to align events by transforming event data into 2D image-like representations.

Photometric consistency between supplementary visual images and event data is guaranteed by event-image alignment techniques. To estimate camera positions and depths, these approaches correlate event data with corresponding pixel brightnesses. Filter-based techniques are employed in direct methods to process incoming event data. For example, one approach uses two filters for camera pose estimation and image gradient calculation under rotational camera motion. The first filter utilizes the current camera pose and Gaussian noise for motion modelling, projecting events to a reconstructed reference image and updating state values based on logarithmic brightness differences. The second filter estimates logarithmic gradients using the linearized Edge Gradient Method (EGM) and employs interleaved Poisson reconstruction for absolute brightness intensity recovery. An alternate method to improve robustness is to estimate additional states for contrast threshold and camera posture history, then filter outliers in event data using a robust sensor model with a normal-uniform mixed distribution.

Several techniques are proposed for estimating camera posture and velocity from event data. One method considers the fact that events are more frequent in areas with large brightness gradients and maximizes a probability distribution function proportional to the magnitude of camera velocity and image gradients. An alternative method makes use of the linearized EGM to determine the

camera motion parameters, calculating both linear and angular velocity by taking the camera's velocity direction into account. Non-linear optimization is used in some techniques to process groupings of events concurrently to reduce the computational cost associated with updating camera positions on an event-by-event basis. These methods estimate camera posture and velocity simultaneously by converting an event stream to a brightness incremental image and aligning it with a reference image. While one approach uses the mapping module's provided photometric 3D map as an assumption, another uses Photometric Bundle Adjustment (PBA) to fine-tune camera positions and 3D structure by transferring depth values between keyframes.

To guarantee photometric consistency, event-image alignment techniques rely on extra information such as brightness pictures and a photometric 3D map with intensities and depths. On the other hand, event representation-based alignment techniques map onto the structure of the frame-based direct method, transforming event data into representations that resemble 2D images. A geometric strategy based on edge patterns is presented by the Event-based VO (EVO) [99] method for aligning event data. It aligns a series of events with the reference frame created by the 3D map's reprojection in its camera tracking module by converting them into an edge map. The mapping module rebuilds a local semi-dense 3D map without explicit data associations using Event-Based Multi-View Stereo (EMVS) [100].

To take advantage of the temporal information contained in event data, Event-Based Stereo Visual Odometry (ESVO) [15] presents an event-event alignment technique on a Time Surface (TS). A TS is interpreted by ESVO as an anisotropic distance field in its camera tracking module, which aligns the support of the semi-dense map with the latest events in the TS. The task of estimating the camera position is expressed as a minimization problem by lining up the support with the negative TS minima. To maximize stereo temporal consistency, ESVO uses a forward-projection technique to reproject reference frame pixels to stereo TS during mapping. By combining the depth distribution in neighborhoods and spreading earlier depth estimates, a depth filter and fusion approach are created to improve the depth estimation. A different approach [18] suggests a selection procedure to help the semi-dense map get rid of unnecessary depth points and cut down on processing overhead. Furthermore, it prevents the degradation of ESVO in scenarios with few generated events by fusing IMU data with the time surface using the IMU pre-integration algorithm [18]. In contrast, Depth-Event Camera Visual Odometry (DEVO) [101] uses a depth sensor to enhance the creation of a precise 3D local map that is less affected by erratic events in the mapping module.

3.2.3. Motion Compensation Methods

Using the event frame as the fundamental event representation, motion-compensation techniques are based on event alignment. To provide clear images and lessen motion blur over a longer temporal window, these algorithms optimize event alignment in the motion-compensated event frame to predict camera motion parameters. On the other hand, there is a chance of unfavorable results, including event collapse, in which a series of events builds up into a line or a point inside the event frame. Contrast Maximization (CMax), Dispersion Minimization (DMin), and Probabilistic Alignment techniques are the three categories into which the approaches are divided.

Using the maximum edge strengths in the Image Warping Error (IWE), the CMax framework [102] aims to align event data caused by the same scene edges. Optimizing the contrast (variance) of the IWE is the next step in the process, which entails warping a series of events into a reference frame using candidate motion parameters. In addition to improving edge strengths, this makes event camera motion estimation easier.

The DMin methods utilize entropy loss on the warped events to minimize the average events dispersion, strengthening edge structures. They do so by warping events into a feature space using the camera motion model. The potential energy and the Sharma-Mittal entropy are used to calculate the entropy loss. The feature vector undergoes a truncated kernel function-based convolution, which leads to a computational complexity that increases linearly with the number of events. Furthermore, an incremental variation of the DMin technique maximizes the measurement function within its spatiotemporal vicinity for every incoming event.

The possibility that event data would correspond to the same scene point is assessed using a probabilistic model that was recently established in [103]. The pixel coordinates of an event stream are rounded to the nearest neighbor using a camera motion model to create a reference timestamp. The Poisson random variable is used to represent the count of warped events at each pixel, while the spatiotemporal Poisson point process (ST-PPP) model is used to represent the probability of all the warped events together. Next, by maximizing the ST-PPP model's probability, the camera motion parameters are approximated.

3.2.4. Deep Learning Methods

Deep learning techniques have been widely used in computer vision applications in recent years, and they have shown a great deal of promise in VSLAM algorithms [104–109]. However, typical Deep Neural Networks (DNNs) including Multi-Layer Perceptron networks (MLPs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs) have difficulties due to the sparse and asynchronous nature of event data collected by event cameras. Currently, available DNNs often require conversion to voxel grids [110] or event-frame-based representations [65] to process event data. Conversely, individual event data can be processed directly and without pre-processing via SNNs. Supervised and unsupervised learning techniques are additional categories of event-based deep learning.

The goal of supervised deep learning techniques is to minimize the discrepancies between the ground truth and the predicted poses and depths. Using a CNN to extract features from the event frame and a stacked spatial Long Short-Term Memory network (LSTM) to merge it with motion history is one method of regressing camera poses from sequences of event data [67]. Nevertheless, this approach has difficulties when it comes to processing collected events and estimating a camera attitude for a subset of event data inside each accumulation of events. Another method for addressing this is a convolutional SNN for preprocessing-free continuous-time camera posture regression [61].

In unsupervised deep learning methods, depth values and ground truth camera postures are not required for training. Rather, they employ supervisory signals, including photometric consistency, which are acquired through the process of back-warping adjacent frames utilizing the depth and pose predictions of DNNs inside the multi-view geometric constraint.

3.3. Performance Evaluation of SLAM Systems

To assess the relative effectiveness of alternative SLAM solutions, reliable evaluation metrics are needed. This section discusses some of the existing metrics and their applicability to event camera-based SLAM implementations.

3.3.1. Event Camera Datasets

The availability of suitable datasets plays a crucial role in testing and validating the performance of novel systems. In this regard, for the evaluation of event camera-based systems, relevant datasets must be prepared from the images or videos captured using an event camera. Neuromorphic vision datasets follow an event-driven processing paradigm represented by binary spikes and have rich spatiotemporal components compared to traditional frame-based datasets [111]. In general, there are two kinds of neuromorphic datasets, DVS-converted (converted from frame-based static image datasets) and DVS-captured datasets [111]. Although DVS-converted (frameless) datasets can contain more temporal information as compared to the original dataset, they come with certain drawbacks (full temporal information cannot be obtained) and are generally considered not to be a good option for benchmarking SNNs [112,113]. Moreover, it has been observed that spike activity decreases in deeper layers of spiking neurons when they are trained on such data, which results in performance degradation during the training [114]. Conversely, DVS-captured datasets generate spike events naturally, which makes it a more suitable sensor input for SNNs [111,114,115].

Several datasets have been developed to facilitate the evaluation of event-based cameras and SLAM systems [19]. The early datasets, such as the one introduced in [116], offer sequences captured

by handheld event cameras in indoor environments, alongside ground truth camera poses obtained from motion capture systems, albeit limited to low-speed camera motions in small-scale indoor settings. Similarly, the RPG dataset [117] also focuses on indoor environments, utilizing handheld stereo event cameras, but is constrained by similar limitations. In contrast, the MVSEC dataset [70] represents a significant advancement, featuring large-scale scenarios captured by a hexacopter and a driving car, encompassing both indoor and outdoor environments with varied lighting conditions. Another notable dataset, the Vicon dataset reported in [118], incorporates event cameras with different resolutions to capture high dynamic range scenarios under challenging lighting conditions. Moreover, recent advancements have led to the release of advanced event-based SLAM datasets [98,119–122] like the UZH-FPV dataset [119], which employs a wide-angle event camera attached to a drone to capture high-speed camera motions in diverse indoor and outdoor environments, and the TUM-VIE dataset [120], which utilizes advanced event cameras to construct stereo visual-inertial datasets spanning various scenarios from small to large-scale scenes with low-light conditions and high dynamic range.

3.3.2. Event-Based SLAM Metrics

In assessing the performance of SLAM algorithms, particularly in terms of camera pose estimation, two primary metrics are commonly utilized: the absolute trajectory error (ATE) and the relative pose error (RPE) [123]. ATE quantifies the accuracy of camera poses relative to a world reference, measuring translational and rotational errors between estimated and ground truth poses. Conversely, RPE evaluates the consistency of relative camera poses between consecutive frames. ATE offers a comprehensive assessment of long-term performance, while RPE provides insights into local consistency. Notably, some studies adjust positional error measurements concerning mean scene depth or total traversed distance for scale invariance [118,124]. Additionally, alternative metrics [110] like Average Relative Pose Error (ARPE), Average Relative Rotation Error (ARRE) and Average Endpoint Error (AEE) are suggested for evaluating translational and rotational differences. ARPE measures the geodesic distance between two rotational matrices, whereas AEE and ARPE quantify the position and orientation differences between two translational vectors, respectively. Average linear and angular velocity errors can also serve as alternative metrics for pose estimation. For depth estimation, the average depth error at various cut-offs up to fixed depth values is commonly employed, allowing for comparisons across diverse scales of 3D maps.

3.3.3. Performance Comparison of SLAM Methods

To evaluate the state-of-the-art methods of SLAM, depth and camera pose estimation quality are additional metrics that can be used to make a performance comparison. In the following section, qualitative analyses based on the existing literature were presented.

3.3.3.1. Depth Estimation

The study [19] assessed three DNN-based monocular depth estimation techniques and compared them to the most advanced conventional approach, which is MegaDepth [71,110], E2Depth [68], and RAM [69]. These techniques were trained using the MVSEC dataset's *outdoor_day 2* sequence [70], and the average depth errors at various maximum cutoff depths (such as 10m, 20m, and 30m respectively) were compared.

According to the results of [19], event-based approaches perform better than frame-based methods when handling fast motion and poor light. MegaDepth's accuracy decreased in nighttime *outdoor_night* sequences taken from moving vehicles because of motion blur and a constrained dynamic range. However, it was discovered that using the reconstructed images made from event streams improved the performance. On average, depth mistakes are regularly 1-2 meters lower with an unsupervised approach [110] than with MegaDepth. The addition of ground truth labels and more training on artificial datasets was found to increase E2Depth's efficacy [19]. Further improvements over these event-based techniques are shown by RAM, which combines synchronous intensity

images with asynchronous event data. This implies that using static features that are taken from intensity images can improve the efficiency of event-based methods.

3.3.3.2. Camera Pose Estimation

Rotating sequences [116] can be used to evaluate motion compensation algorithms by measuring the Root Mean Square (RMS) of the angular velocity errors. With the least amount of temporal complexity among the assessed techniques, CMax [102] was discovered to exhibit good performance for the 3-DoF rotational motion of event cameras. With the addition of entropy minimization on projected events, DMin [125] improves CMax's performance in high-dimensional feature spaces by about 20%. However, DMin comes at a significant computational expense. This problem was addressed by Approximate DMin [125], which uses a shortened kernel for increased efficiency. With a 39% improvement in the shape sequence, an alternate method using a probabilistic model, ST-PPP [103], achieved the best performance of all the methods studied.

To assess the performance of both motion-compensation and deep learning techniques on the *outdoor day 1* sequence in [110], metrics such as ARPE, ARRE and AEE were used. It was discovered that DMin [126] performs best when the dispersion of back-projected events in 3D space is kept to a minimum. Additionally, Approximate DMin has reduced the time complexity and outperformed the standard DMin by about 20%. However, the online version of DMin has produced inferior results because of its event-by-event processing. It was discovered that deep-learning techniques outperformed motion-compensation techniques [65].

Research has employed *boxes* [116] and *pipe* [124] sequences to measure positional mistakes with mean scene depth and orientation errors, to compare the two event-image alignment techniques. Utilizing a filter-based approach that takes advantage of the photometric link between brightness change and absolute brightness, [124] demonstrated very good results. On the other hand, [127] aligns two brightness incremental photos using least-squares optimization to produce even better results.

The RPG dataset [15] has been used to evaluate several EVO algorithms with respect to positional and orientation errors. EVO [99] performed well in a variety of sequences, but it had trouble keeping up with abrupt changes in edge patterns. Outperforming EVO, Ultimate SLAM (USLAM) [79] improved feature-based VO by fusing pictures and inertial data with event data. When it comes to camera pose estimation, ESVO [15] outperformed USLAM and provided more accurate depth estimation from stereo event cameras, however, it still lagged behind frame-based algorithms like Direct Sparse Odometry (DSO) [128] and ORB-SLAM2 [56]. By using photometric bundle correction, Event-aided DSO (EDSO) [98] attained performance that is equivalent to DSO. Additionally when the reconstructed images from E2VID [129] are taken as an input, DSO achieved better performance in the *rpg_desk* sequence. Nevertheless, DSO has trouble with high-texture sequences because of E2VID reconstruction problems.

Additionally, the assessment of many EVIO techniques was done using the VICON dataset [118], emphasizing positional inaccuracies in relation to the ground truth trajectory's overall trajectory length. When it comes to combining event data with IMU data and intensity images, USLAM underperformed the frame-based VIO algorithms (SOTA) [81]. With event-corner feature extraction, tracking methods, and sliding-windows graph-based optimization, EIO [130] improves performance. Additionally, PL-EVIO [118] outperformed both event-based and frame-based VIO techniques by extending line-based features in event data and point-based features in intensity images.

3.4. Applications of Event Camera-Based SLAM Systems

Due to their unique advantages, event cameras are gaining increasing attention in various fields, including robotics and computer vision. The utilization of event cameras in the SLAM field has the potential to enable several valuable applications in a variety of fields, as discussed below.

3.4.1. Robotics

Event-based SLAM systems have the transformative potential to empower robots with autonomous navigation capabilities even in the most challenging and cluttered environments. By leveraging the asynchronous and high-temporal-resolution data provided by event-based cameras, these systems can offer robots a nuanced understanding of their surroundings, enabling them to navigate with significantly improved precision and efficiency. Unlike traditional SLAM methods, event-based SLAMs excel in capturing rapid changes in the environment, allowing robots to adapt swiftly to dynamic obstacles and unpredictable scenarios. This heightened awareness not only enhances the safety and reliability of robotic navigation but also opens doors to previously inaccessible environments where real-time responsiveness is paramount.

Obstacle avoidance represents a critical capability in the realm of robotic navigation, and event-based cameras offer potential advantages for the real-time perception of dynamic obstacles. Event-based sensors will enable robots to swiftly detect and respond to changes in their environment, facilitating safe traversal through complex and challenging landscapes. By continuously monitoring their surroundings with a high temporal resolution, event-based cameras can enable robots to navigate complex dynamic environments, avoiding collisions and hazards in real-time. This capability would not only enhance the safety of robotic operations in dynamic environments but also unlock new possibilities for autonomous systems to be integrated into human-centric spaces, such as high-traffic streets or crowded indoor environments.

Event-based SLAM systems also provide advantages for tracking moving objects in various critical applications. The ability to monitor and follow dynamic entities is important in many applications including navigation in dynamic environments, or object manipulation tasks. Event-based cameras, due to their rapid response times and precise detection capabilities, can theoretically be used to capture the motion of objects accurately and efficiently. This real-time tracking functionality will not only enhance situational awareness capability but also facilitate timely autonomous decision-making processes in dynamic and time-sensitive scenarios.

3.4.2. Autonomous Vehicles

The integration of event-based SLAM systems can provide benefits in the realm of self-driving cars. The unique characteristics of event-based cameras with regards to high temporal resolution and adaptability to dynamic lighting conditions, in conjunction with other sensors, could provide autonomous vehicles with improved capability to navigate through challenging scenarios such as in low light or during adverse weather conditions.

Effective collision avoidance systems are vital for the safe operation of autonomous vehicle technology, and the integration of event-based cameras has the potential to enhance these systems. By leveraging the unique capabilities of event-based cameras, autonomous vehicles can achieve real-time detection and tracking of moving objects with high levels of precision and responsiveness. By providing high-temporal-resolution data, event-based cameras offer a granular understanding of dynamic traffic scenarios, potentially improving the ability of vehicles to avoid hazardous situations.

3.4.3. Virtual Reality (VR) and Augmented Reality (AR)

With their high temporal resolution and low latency, event camera-based SLAM systems could provide advantages for the accurate inside-out real-time tracking of head movements or hand gestures, which are important capabilities for immersive VR systems. Their low power requirements would also provide significant benefits for wireless headsets.

Event-based SLAM systems could also provide advantages in the realm of spatial mapping, particularly for Augmented Reality (AR) applications. Their ability to capture changes in the environment with high temporal resolution, and with robustness to variations in lighting, should enable event-based cameras to create accurate spatial maps in a variety of conditions.

4. Application of Neuromorphic Computing to SLAM

Machine learning algorithms have become more powerful and have shown success in various scientific and industrial applications due to the development of increasingly powerful computers and smart systems. Influenced by the hierarchical nature of the human visual system, deep-learning techniques have undergone remarkable advancement [131]. Even with these developments, the mainstream machine learning (ML) models in robotics can still not perform tasks with human-like ability, especially in tasks requiring fine motor control, quick reflexes, and flexibility in response to changing environments. There are also scalability issues with these standard machine-learning models.

The difference in power consumption between the human brain and current technology is striking when one realizes that a clock-based computer operating a "human-scale" brain simulation in theory would need about 12 gigawatts, but the human brain only uses 20 Watts [132]. The artificial discretization of time imposed by mainstream processing and sensor architectures [133], which depend on arbitrary internal clocks, is a major barrier to the upscaling of intelligent interactive agents. To process the constant inputs from the outside world, clock frequencies must be raised. However, with present hardware, obtaining such high frequencies is not efficient and practicable for large-scale applications. Biological entities use spikes for information processing to digest information at a high rate of efficiency, which improves their perception and interaction with the outside world. In the quest for computer intelligence that is comparable to that of humans, one difficulty is to replicate the effective neuro-synaptic architecture of the physical brain. Several technologies and techniques aimed at more accurately mimicking the biological behavior of the human brain have been developed because of the considerable exploration of this area in recent years. This conduct is marked by quick response times and low energy use. Neuromorphic computing, sometimes referred to as brain-inspired computing, is one notable strategy in this quest.

A multidisciplinary research paradigm called "neuromorphic computing" investigates large-scale processing devices that use spike-driven communication to mimic natural neural computations. When compared to traditional methods, it has several advantages, such as energy efficiency, quick execution, and robustness to local failures [134]. Moreover, the neuromorphic architecture employs asynchronous event-driven computing to mitigate the difficulties associated with the artificial discretization of time. This methodology is consistent with the external world's temporal progression. Inspired by this event-driven information processing, advances in neuroscience and electronics, in both hardware and software, have made it possible to design systems that are biologically inspired. SNNs are used in these systems to simulate interactive and cognitive functions [135].

In the discipline of neurorobotics, which includes both robotics and neuromorphic computing, bio-inspired sensors are essential for efficiently encoding sensory inputs. Furthermore, these sensors combine inputs from many sources and use event-based computation to accomplish tasks to adjust to different environmental conditions [136]. To date, however, not much study has been focused on the application of neuromorphic computing in SLAM, even though various companies have built experimental neuromorphic processors in the last ten or so years. This is primarily because practical implementations are only now beginning to become accessible.

4.1. Neuromorphic Computing Principles

The development of neuromorphic hardware strives to provide scalable, highly parallel, and energy-efficient computing systems. These designs are ideal for robotic applications where rapid decision-making and low power consumption are critical since they are made to process data in real time with low latency and high accuracy. Because they require vast volumes of real-time data processing, certain robotics tasks, such as visual perception and sensor fusion, are difficult for ordinary CPUs/GPUs to handle. For these kinds of activities, traditional computing architectures, such as GPUs, can be computationally and energy-intensive. By utilizing the distributed and parallel characteristics of neural processing, neuromorphic electronics offer a solution and enable effective real-time processing of sensory data. Furthermore, conventional computing architectures do poorly on tasks requiring cognitive capacities like those of humans, such as learning, adapting, and making

decisions, especially when the input space is poorly defined. In contrast, they perform exceptionally well on highly structured tasks like arithmetic computations [25].

Neuromorphic Computers consist of neurons and synapses rather than a separate central processing unit (CPU) and memory units [26]. As their structure has gained inspiration from the working of the biological brain, the structure and function are similar to the brain where neurons and synapses are responsible for processing and memory respectively [26]. Moreover, neuromorphic systems natively take inputs as spikes (rather than binary values) and these spikes generate the required output. The challenge to realizing the true potential of neuromorphic hardware lies with the development of a reliable computing framework that enables the programming of the complete capabilities of neurons and synapses in hardware as well as methods to communicate effectively between neurons to address the specified problems [27,28].

The advent of neuromorphic processors that employ various sets of signals to mimic the behavior of biological neurons and synapses [11,27,28] has paved a new direction in the neuroscience field. This enables the hardware to asynchronously communicate between its components and the memory in an efficient manner, which results in lower power consumption in addition to other advantages [11,26,28]. These neuromorphic systems are fully event-driven and highly parallel in contrast to traditional computing systems [26]. Today's von Neumann CPU architectures and GPU variations adequately support Artificial Neural Networks (ANNs), particularly when supplemented by coprocessors optimized for streaming matrix arithmetic. These conventional architectures are, however, notably inefficient in catering to the needs of SNN models [137].

In addition, as computation in neuromorphic systems is fundamentally based on neural networks, neuromorphic computers become a highly relevant platform for use in artificial intelligence and machine learning applications to enhance robustness and performance [26,29]. This has encouraged and attracted researchers [28,32,133,138–140] to further explore applications and development. The development of SpiNNaker (Spiking Neural Network Architecture) [141,142] and BrainScaleS [143,144] was sponsored by the European Union’s Human Brain Project to be used in the neuroscience field. Similarly, developments such as IBM’s TrueNorth [145], Intel’s Loihi [28,137] and Brainchip’s Akida [146] are some of the indications of success in neuromorphic hardware development [147].

In the following sections, recent neuromorphic developments are identified and described. Table 2 gives a summary of the currently available neuromorphic processing systems.

Table 2. Comparison of neuromorphic processor architectures.

Year	Processor/ Chips	I/O	On-device Training	Event- based	Feature Size (nm)	Power
2011	SpiNNaker	Real Numbers, Spikes	STDP	No	22	20 nj/operation
2014	TrueNorth	Spikes	No	Yes	28	0.18W
2018	Loihi	Spikes	STDP	Yes	14	80 pj /operation
2020	BrainScaleS	Real Numbers, Spikes	STDP, Surrogate Gradient	Yes	65	0.2W
2021	Loihi2	Real Numbers, Spikes	STDP, Surrogate, Backpropagation	Yes	7	-
2021	DYNAP SE2, SEL, CNN	Spikes	STDP (SEL)	Yes	22	10mW
2021	Akida	Spikes	STDP (Last Layer)	Yes	28	100 μW– 300 mW

4.1.1. SpiNNaker

The University of Manchester's SpiNNaker project launched the first hardware platform designed specifically for SNN research in 2011 [141]. A highly parallel computer was created, SpiNNaker 2 [148] in 2018 as a part of the European Human Brain Project. Its main component is a

specially made micro-circuit with 144 ARM M4 microprocessors and 18 Mbyte of SRAM. It has a limited instruction set but performs well and uses little power. Support for rate-based DNNs, specialized accelerators for numerical operations, and dynamic power management are just a few of the new features that SpiNNaker 2 offers [149].

The SpiNNaker chips are mounted on boards, with 56 chips on each board. These boards are then assembled into racks and cabinets to create the SpiNNaker neurocomputer, which has 106 processors [150]. The system functions asynchronously, providing flexibility and scalability; however, it requires the use of AER packets for spike representation through the implementation of multiple communication mechanisms.

Researchers can more successfully mimic biological brain structures with the help of SpiNNaker. It was noteworthy that it outperformed GPU-based simulations in real-time simulation for a 1 mm² cortical column (containing 285,000,000 synapses and 77,000 neurons at a 0.1 ms time-step) [151]. SpiNNaker's intrinsic asynchrony makes it easier to represent a 100 mm² column by increasing the number of computing modules; a task that GPUs find challenging because of synchronization constraints.

4.1.2. TrueNorth

In 2014, IBM launched the TrueNorth project, the first industrial neuromorphic device, as part of DARPA's SyNAPSE programme [152]. With 4,096 neural cores that can individually simulate 256 spiking neurons in real time, this digital device has about 100 Kbits of SRAM memory for storing synaptic states. Using a digital data highway for communication, neurons encode spikes as AER packets. TrueNorth neural cores can only perform addition and subtraction; they cannot perform multiplication or division, and their functionality is fixed at the hardware level [149].

There are 256 common inputs in each neural core, which enables arbitrary connections to the 256 neurons inside the core. Because synapse weights are only encoded with two bits, learning methods cannot be implemented entirely on the chip. For running recurrent (RNN) and convolutional neural networks (CNN) in inference mode, True-North is a good choice [152]. But to transfer learnt weights into TrueNorth configurations for learning, an extra hardware platform typically requires a GPU.

An example application from 2017 [23] uses a TrueNorth chip and DVS camera to create an event-based gesture detection system. It took 0.18 W and 0.1 seconds to recognize 10 gestures with 96.5% accuracy. The same researchers demonstrated an event-based stereo-vision system in 2018 [153] that boasted 200 times more energy economy than competing solutions. It used two DVS cameras and eight True-North CPUs, and it could determine scene depth at 2,000 disparity maps per second. Furthermore, in 2019, a scene-understanding application showed how to detect and classify several objects at a throughput of more than 100 frames per second from high-definition aerial video footage [145].

4.1.3. Loihi

The first neuromorphic microprocessor with on-chip learning capabilities was introduced in 2018 with the release of Intel's Loihi project [137]. Three Pentium processors, four communication modules, and 128 neural cores are all integrated into a single Loihi device to enable the exchange of AER packets. With 128 Kbytes of SRAM for synapse state storage, each of these cores may individually simulate up to 1,024 spiking neurons. The chip can simulate up to 128,000,000 synapses and about 128,000 neurons in this setup. The mechanism maintains spike transmission from neuron to neuron smoothly and modifies its speed if the spike flow gets too strong.

Loihi allows for on-chip learning by dynamically adjusting its synaptic weights, which range from 1 to 9 bits [149]. A variable that occupies up to 8 bits and acts as an auxiliary variable in the plasticity law is included in each synapse's state, along with a synaptic delay of up to 6 bits. Only addition and multiplication operations are required for local learning, which is achieved by weight recalculation during core configuration.

Various neurocomputers have been developed using Loihi, with Pohoiki Springs being the most potent, combining 768 Loihi chips into 24 modules to simulate 100,000,000 neurons [149]. Loihi is

globally employed by numerous scientific groups for tasks like image and smell recognition, data sequence processing, PID controller realization, and graph pathfinding [28]. It is also utilized in projects focusing on robotic arm control [154] and quadcopter balancing [155].

With 128 neural cores that can simulate 120,000,000 synapses and 1,000,000 programmable neurons, Intel unveiled Loihi 2, a second version, in 2021 [28]. It integrates 3D multi-chip scaling, which enables the combining of numerous chips in a 3D environment and makes use of Intel's 7nm technology for a 2.3 billion transistor chip. With local broadcasts and graded spikes where spike values are coded by up to 32 bits; Loihi 2 presents a generalized event-based communication model. An innovative approach to process-based computing was presented by Intel with the launch of the Lava framework [156], an open-source platform that supports Loihi 2 implementations on CPU, GPU, and other platforms [28].

4.1.4. BrainScaleS

As part of the European Human Brain Project, Heidelberg University initiated the BrainScaleS project in 2020 [157]. Its goal is to create an Application-Specific Integrated Circuit (ASIC) that can simulate spiking neurons by using analog computations. Analog computations are performed using electronic circuits, which are characterized by differential equations that mimic the activity of organic neurons. Every electronic circuit consists of a resistor and a capacitor, symbolizing a biological neuron. The second version of the 2011 release had digital processors to facilitate local learning (STDP) in addition to the analog neurons, whereas the first version did not include on-chip learning capabilities [149]. Spikes in the form of AER packets are used as a digital data highway to promote communication between neurons. 130,000 synapses and 512 neurons can be simulated on a single chip.

While the analog neuron model has advantages over biological neurons (up to 10,000 times faster in analog implementation) and adaptability (compatible with classical ANNs) [158], it also has drawbacks due to its relative bulk and lack of flexibility [149]. BrainScaleS has been used to tackle tasks in a variety of fields, such as challenges involving ANNs, speech recognition utilizing SNNs, and handwritten digit recognition (MNIST) [32,159–161]. For example, BrainScaleS obtained a 97.2% classification accuracy with low latency, energy consumption, and total chip connections using the spiking MNIST dataset [149]. To implement on-chip learning, surrogate gradient techniques were used [162].

The use of BrainScaleS for Reinforcement Learning tasks using the R-STDP algorithm demonstrated the platform's potential for local learning [163]. An Atari PingPong-like computer game was used to teach the system how to manipulate a slider bar [149].

4.1.5. Dynamic Neuromorphic Asynchronous Processors

A group of neuromorphic systems called Dynamic Neuromorphic Asynchronous Processors (DYNAP) were created by SynSense, a University of Zurich affiliate, using patented event-routing technology for core communication. A significant barrier to the scalability [164] of neuromorphic systems is addressed by SynSense's unique two-level communication model, which optimizes the ratio of broadcast messages to point-to-point communication inside neuron clusters. The research chips DYNAP-SE2 and DYNAP-SEL are part of the DYNAP family and are intended for use by neuroscientists investigating SNN topologies and communication models. Furthermore, there is DYNAP-CNN, a commercial chip designed specifically to efficiently perform SNNs that have been converted from CNNs. Analog processing and digital communication are used by DYNAP-SE2 and DYNAP-SEL, whilst DYNAP-CNN is entirely digital, enabling event-based sensors (DVS) and handling image classification tasks.

DYNAP-SE2 has four cores with 65k synapses and 1k Leaky Integrate-and-Fire with Adaptive Threshold (LIFAT) analog spiking neurons, making it suitable for feed-forward, recurrent, and reservoir networks [149]. This chip, which offers many synapse configurations (N-methyl-D-aspartate (NMDA), α -amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid (AMPA), Gamma-aminobutyric acid type A (GABA_A), and Gamma-aminobutyric acid type B (GABA_B)), makes

research into SNN topologies and communication models easier. With five cores, including one with plastic synapses, DYNAP-SEL has a huge fan in/out network connectivity and facilitates on-chip learning. Researchers can mimic brain networks with the chip's 1,000 analog spiking neurons and up to 80,000 reconfigurable synaptic connections, of which 8,000 include spike-based learning rules (STDP).

The Dynap-CNN chip has been available with a Development Kit since 2021. It is a 12 mm² chip with four million configurable parameters and over a million spiking neurons, built using 22nm technology. It only runs in the inference mode and performs effective SNN conversion from CNNs. It achieves notable performance on applications including wake phrase identification, attentiveness detection, gesture recognition, and CIFAR-10 picture classification. There is no support for on-chip learning; the initial CNN needs to be trained on a GPU using traditional techniques like PyTorch and then converted using the Sinabs.ai framework so that it can run on Dynap-CNN.

4.1.6. Akida

Akida, created by Australian company BrainChip, stands out as the first commercially available neuromorphic processor released in August 2021 [165], with NASA and other companies participating in the early access program. Positioned as a power-efficient event-based processor for edge computing, Akida functions independently of an external CPU and consumes 100 μ W to 300 mW for diverse tasks. Boasting a processing capability of 1,000 frames/Watt, Akida currently supports convolutional and fully connected networks, with potential future backing for various neural network types. The chip facilitates the conversion of ANN networks into SNN for execution.

A solitary Akida chip within a mesh network incorporates 80 Neural Processing Units, simulating 1,200,000 neurons and 10,000,000,000 synapses. Fabricated using TSMC technology, a second-generation 16 nm chip was unveiled in 2022. The Akida ecosystem encompasses a free chip emulator, the MetaTF framework for network transformation, and pre-trained models. Designing for Akida necessitates consideration of layer parameter limitations.

A notable feature of Akida is its on-chip support for incremental, one-shot, and continuous learning. BrainChip showcased applications at the AI Hardware Summit 2021, highlighting human identification after a single encounter and a smart speaker using local training for voice recognition. The proprietary homeostatic STDP algorithm supports learning, with synaptic plasticity limited to the last fully connected layer. Another demonstrated application involved the classification of fast-moving objects using an event-based approach, effectively detecting objects even when positioned off-centre and appearing blurred.

4.2. Spiking Neural Networks

Typically, neural networks are divided into three generations, each of which mimics the multilayered structure of the human brain while displaying unique behaviors [166]. The first generation has binary (0,1) neuron output, which is derived from simple weighted synaptic input thresholding. The research [167] showed that networks made of artificial neurons could perform logical and mathematical operations. With the advancement of multi-layer perceptron networks and the backpropagation technique, a new idea became apparent over time. In modern deep learning, this method is commonly used to overcome the shortcomings of earlier neural perceptron techniques. Artificial Neural Networks are the name given to this second generation (ANNs). Its primary distinction from the first generation lies in neuron output, which can be a real number resulting from the weighted sum of inputs processed through a transfer function, typically sigmoidal. Weights are determined through various machine-learning algorithms, ranging from basic linear regression to advanced classification.

Compared to their biological counterparts, neural networks in their first and second generations have limited modelling capabilities. Interestingly, there is no temporal reference to electrical impulses found in biological neural networks in these models. Additionally, research on biological processes remains limited. The human brain excels in processing real-time data, efficiently encoding information through various features related to spikes, including specific event times [168]. The

concept of simulating neural events prompted the creation of SNNs, which currently stand as the most biologically plausible models.

An SNN architecture controls the information transfer from a presynaptic (source) neuron to a postsynaptic (target) neuron through a network of interconnected neurons connected by synapses. SNNs use spikes to encode and transport information, in contrast to traditional ANNs. Unlike a single forward propagation, each input is displayed for a predefined duration (T), resulting in several forward passes, $\frac{T}{\delta t}$. Like the biological counterpart, a presynaptic neuron that fires transmit a signal proportionate to the synapse weight or conductance to its postsynaptic counterpart in the form of a synaptic current. Generally, when the synaptic current enters the target neuron, it causes a certain amount, δv , to change in the membrane potential (v_{mem}). The postsynaptic neuron fires a spike and resets its membrane voltage to the resting potential (v_{rest}) if the v_{mem} crosses a predetermined threshold (v_{thresh}). On the other hand, different network topologies and applications may require different combinations of learning rules, v_{mem} dynamics, and neuron models. Various methodologies can be used to describe neurons and synapse dynamics.

Compared to standard ANNs, SNNs include topologies and learning principles that closely mimic biological processes. SNNs, the third generation of ANNs, are excellent at reliable computation with little computational load. SNN neurons are not differentiable; once their states cross the firing threshold, they produce event-based spikes, but they also hold onto past states that gradually deteriorate over time. Because SNNs are dynamic, direct training with the conventional backpropagation (BP) method is difficult and considered biologically unrealistic [169]. To substitute ReLU activation functions in the ANN with the Leaky Integrate and Fire (LIF) neurons, SNNs are thus created from trained ANNs [170]. However, converted SNNs generally fail to achieve the required performance and impact the latency and power consumption. This has led to directly training SNNs using both unsupervised STDP and supervised learning rules (such as SpikeProp and gradient-based learning) that have also resulted in producing inefficient results, but the surrogate gradient learning rule was found to be effective in training complex and powerful SNNs [170].

Among the various neuron models proposed by the researchers, the LIF and its variants are among the most popular neuron models due to their low implementation costs [29]. The LIF model can be represented mathematically as:

$$C \frac{dV}{dt} = -g_L(V(t) - E_L) + I(t) \quad (3)$$

In equation 3, output voltage $V(t)$ relies on the conductance g_L of the resistor, the capacitance C of the capacitor, the resting voltage E_L , and a current source $I(t)$. When multiplying equation 3 by $R = \frac{1}{C}$, the $\frac{dv_{mem}}{dt}$ in relation to the membrane time constant, τ_m is:

$$\tau_m \frac{dv_{mem}}{dt} = -[v_{mem}(t) - v_{rest}] + RI \quad (4)$$

Consequently, the activation function $A(t)$ for LIF neurons is represented as shown in 5; v_{mem} constantly decays to the rest value and undergoes a refractory period.

$$A(t) = \begin{cases} 0, & \text{if } v_{mem} < v_{thresh} \\ 1, & \text{if } v_{mem} \geq v_{thresh} \end{cases} \quad (5)$$

4.3. Neuromorphic Computing in SLAM

Integrating neuromorphic computing into SLAM systems involves merging the distinctive characteristics of neuromorphic hardware and algorithms to enhance SLAM's functionality. The integration needs to leverage the unique capabilities of neuromorphic computing to improve the performance and efficiency of SLAM operations [171–173]. This requires adapting and utilizing neuromorphic technology to address various aspects of SLAM, from sensor data processing to navigation and planning. To date, the application of neuromorphic computing in SLAM technology has had limited exploration, but its use in other related areas has been widely studied [28,172,174–176].

In medical treatment and monitoring, neuromorphic systems are worn or implanted as components of other medical treatment tools or interface directly with biological systems [32,177]. The concept of neuromorphic brain-machine interfaces [178–183] have become popular as communication between machine and brain can happen naturally with the biological and neuromorphic systems both following spike-based communications. Similarly, in robotics, where on-board processing needs to be very compact and power-efficient, the most common existing applications of neuromorphic systems include behavior learning, locomotion control, social learning, and target learning. However, autonomous navigation tasks are the most common neuromorphic implementations in robotics [32].

Neuromorphic systems have also been used in a wide range of control applications [184–187] because these usually have strict real-time performance requirements, are often used in real systems that have low power and small volume requirements and frequently involve temporal processing, which makes models that use recurrent connections or delays on synapses beneficial [32]. However, the cart-pole problem, also known as the inverted pendulum task [188,189], is the most used control test case. Additionally, neuromorphic systems have been used in video games, including Flappy Bird [190], PARTitioning and Configuration MANager (PACMAN) [191], and Pong [192].

Neural network and neuromorphic implementations have been widely applied to a variety of image-based applications, such as feature extraction [193–195], edge detection [196,197], segmentation [198,199], compression [200,201], and filtering [202,203]. Applications such as image classification, detection, or identification are also very common [32]. Furthermore, applications of neuromorphic systems have also included the recognition of other patterns, such as pixel patterns or simple shapes [204,205]. Additionally, general character identification tasks [206–208] and other digit recognition tasks [209–211] have become highly popular. To assess the numerous neuromorphic implementations, the MNIST data set and its variations have been employed [32,159–161].

Additional image classification tasks have been demonstrated on neuromorphic systems, which include the classifying of real-world images such as traffic signs [212–215], face recognition or detection [216–221], car recognition or detection [221–225], identifying air pollution in images [221,226,227], identifying manufacturing defects or defaults [228,229], hand gesture recognition [221,230,231], object texture analysis [232,233], and other real-world image recognition tasks [221,234]. The employment of neuromorphic systems in video-based applications has also been common [32]; video frames are analyzed as images and object recognition is done without necessarily taking into consideration the time component [235–238]. Nevertheless, a temporal component is necessary for several additional video applications, and further works have investigated this for applications such as activity recognition [239–241], motion tracking [242,243], motion estimation [244–246] and motion detection [243].

In general, the application of neuromorphic systems has been commonly explored in the aforementioned fields as it is found to bring improvement in energy efficiency and performance as compared to a traditional computing platform [171–173,247]. This has led researchers to explore the incorporation of neuromorphic systems in some of the SLAM implementations such as in [24,247–251], resulting in enhanced energy efficiency and performance in addition to other benefits. In [24], when the system (which represented the robot's 4DoF pose in a 3D environment) was integrated with a lightweight vision system (in a similar manner to the vision system of mammalian), the system could generate comprehensive 3D experience maps with consistency both for simulated and real 3D environments. Using the self-learning hardware architecture (gated-memristive device) in conjunction with the spiking neurons, the SLAM system was successful in making navigation-related operations in a simple environment consuming minimal power (36 μ W) [248]. Similarly, the research by [247,250] has shown that power consumption is minimal when the system employs the pose-cell array and digital head direction cell, which mimics place and head direction cells of the rodent brain respectively. Also, the paper on Multi-Agent NeuroSLAM (MAN-SLAM) [249] when coupled with time-domain SNN-based pose cells was found to address the issue of [24] and also improve the accuracy of SLAM results. In a similar way, the methods reported in [251] based on ORB features combined with head direction cells and 3D grid cells was found to enhance the robustness, mapping

accuracy and efficiency of storage and operation. However, the full potential of the technology could not be realized in these studies as novel training and learning algorithms are required to be designed specifically to support the emerging neuromorphic processing technologies, and further work is required to develop these [32,173,176].

These previous examples demonstrate the feasibility and advantages of applying neuromorphic approaches to a range of dynamic modelling and image analysis problems, which provides strong support for the idea that integration of neuromorphic computing into SLAM systems holds significant promise for developing more capable, adaptive, and energy-efficient robotic platforms [171–173,247]. By leveraging the power of neuromorphic hardware and algorithms, SLAM systems should achieve enhanced performance, robustness, and scalability, paving the way for a new generation of intelligent robotic systems capable of navigating and mapping complex environments with increased efficiency and accuracy [28,172,174–176].

Based on this review, it is apparent that successful integration of neuromorphic processing with an event camera-based SLAM system has the potential to provide a number of benefits including:

- **Efficiency:** Neuromorphic hardware is designed to mimic the brain's parallel processing capabilities, resulting in efficient computation with low power consumption. This efficiency is particularly beneficial in real-time SLAM applications where rapid low-power processing of sensor data is crucial.
- **Adaptability:** Neuromorphic systems can adapt and learn from their environment, making them well-suited for SLAM tasks in dynamic or changing environments. They can continuously update their internal models based on new sensory information, leading to improved accuracy and robustness over time.
- **Event-based Processing:** Event cameras capture data asynchronously in response to changes in the environment. This event-based processing enables SLAM systems to focus computational resources on relevant information, leading to faster and more efficient processing compared to traditional frame-based approaches.
- **Sparse Representation:** Neuromorphic algorithms can generate sparse representations of the environment, reducing memory and computational requirements. This is advantageous in resource-constrained SLAM applications, such as those deployed on embedded or mobile devices.

While neuromorphic computing holds promise for enhancing SLAM capabilities [171–173], several challenges will need to be overcome to fully exploit its potential in real-world applications [32,173,176]. Collaboration between researchers in neuromorphic computing, robotics, and computer vision will be crucial in addressing these challenges and realizing the benefits of neuromorphic SLAM systems. One challenge is that neuromorphic hardware is still in the early stages of development, and integration of neuromorphic computing into SLAM systems may require custom hardware development or significant software adaptations. More significantly, adapting existing SLAM algorithms for implementation on neuromorphic hardware is a complex task that requires high levels of expertise in robotics and neuromorphic systems [32,173,176]. Significant research effort will be required to develop and refine these neuromorphic algorithms before outcomes of a comparable level to current state-of-the-art SLAM systems can be achieved.

5. Conclusion

SLAM based on event cameras and neuromorphic computing represents an innovative approach to spatial perception and mapping in dynamic environments. Event cameras capture visual information asynchronously, responding immediately to changes in the scene with high temporal resolution. Neuromorphic computing, inspired by the brain's processing principles, has the capacity to efficiently handle this event-based data, enabling real-time, low-power computation. By combining event cameras and neuromorphic processing, SLAM systems could achieve several advantages, including low latency, low power consumption, robustness to changing lighting conditions, and adaptability to dynamic environments. This integrated approach offers efficient, scalable, and robust solutions for applications such as robotics, augmented reality, and autonomous

vehicles, with the potential to transform spatial perception and navigation capabilities in various domains.

5.1. Summary of Key Findings

VSLAM systems based on traditional image sensors such as monocular, stereo or RGB-D cameras have gained significant development attention in recent decades. These sensors can gather detailed data about the scene and are available at affordable prices. They also have relatively low power requirements, making them feasible for autonomous systems such as self-driving cars, unmanned aerial vehicles and other mobile robots. VSLAM systems employing these sensors have achieved reasonable performance and accuracy but have often struggled in real-world contexts due to high computational demands, limited adaptability to dynamic environments, and susceptibility to motion blur and lighting changes. Moreover, they face difficulties in real-time processing, especially in resource-constrained settings like autonomous drones or mobile robots.

To overcome the drawbacks of these conventional sensors, event cameras have begun to be explored. They have been inspired by the working of biological retinas and attempt to mimic the characteristics of human eyes. This biological design influence for event cameras means they consume minimal power and operate with lower bandwidth in addition to other notable features such as very low latency, high temporal resolution, and wide dynamic range. These attractive features make event cameras highly suitable for robotics, autonomous vehicles, drone navigation, and high-speed tracking applications. However, they operate on a fundamentally different principle compared to traditional cameras; event cameras respond to the brightness changes of the scene and generate events rather than capturing the full frame at a time. This poses challenges as algorithms and approaches employed in conventional image processing and SLAM systems cannot be directly applied and novel methods are required to realize their potential.

For SLAM systems based on event cameras, relevant methods can be selected based on the event representations and the hardware platform being used. Commonly employed methods for event-based SLAM are feature-based, direct, motion-compensated and deep-learning. Feature-based methods can be computationally efficient as they only deal with the small numbers of events produced by the fast-moving cameras for processing. However, their efficacy diminishes while dealing with a texture-less environment. On the other hand, the direct method can achieve robustness in a texture-less environment, but it can only be employed for moderate camera motions. Motion-compensated methods can offer robustness in high-speed motion as well as in large-scale settings, but they can only be employed for rotational camera motions. Deep learning methods can be effectively used to acquire the required attributes of the event data and generate the map while being robust to noise and outliers. However, this requires large amounts of training data, and performance cannot be guaranteed for different environment settings. SNNs have emerged in recent years as alternatives to CNNs and are considered well-suited for data generated by event cameras. The development of practical SNN-based systems is, however, still in the early stages and relevant methods and techniques need considerable further development before they can be implemented in an event camera-based SLAM system.

For conventional SLAM systems, traditional computing platforms usually require additional hardware such as GPU co-processors to perform the heavy computational loads, particularly when deep learning methods are employed. This high computational requirement means power requirements are also high, making them impractical for deployment in mobile autonomous systems. However, neuromorphic event-driven processors utilizing SNNs to model cognitive and interaction capabilities show promise in providing a solution. The research on implementing and integrating these emerging technologies is still in the early stages, however, an additional research effort will be required to realize this potential.

This review has identified that a system based on event cameras and neuromorphic processing presents a promising pathway for enhancing state-of-the-art solutions in SLAM. The unique features of event cameras, such as adaptability to changing lighting conditions, support for high dynamic range and lower power consumption due to the asynchronous nature of event data generation are

the driving factors that can help to enhance the performance of the SLAM system. In addition, neuromorphic processors, which are designed to efficiently process and support parallel incoming event streams, can help to minimize the computational cost and increase the efficiency of the system. Such a neuromorphic SLAM system has the possibility of overcoming significant obstacles in autonomous navigation, such as the need for quick and precise perception, while simultaneously reducing problems relating to real-time processing requirements and energy usage. Moreover, if appropriate algorithms and methods can be developed, this technology has the potential to transform the realm of mobile autonomous systems by enhancing their agility, energy efficiency, and ability to function in a variety of complex and unpredictable situations.

5.2. Current State-of-the-Art and Future Scope

During the last few decades, much research has focused on implementing SLAM based on frame-based cameras and laser scanners. Nonetheless, a fully reliable and adaptable solution has yet to be discovered due to the computational complexities and sensor limitations, leading to systems requiring high power consumption and having difficulty adapting to changes in the environment, rendering them impractical for many use cases, particularly for mobile autonomous systems. For this reason, researchers have begun to shift focus to finding alternative or new solutions to address these problems. One promising direction for further exploration was found to be the combination of an event camera and neuromorphic computing technology due to the unique benefits that these complementary approaches can bring to the SLAM problem.

The research to incorporate event cameras and neuromorphic computing technology into a functional SLAM system is, however, currently in the early stages. Given that the algorithms and techniques employed in conventional SLAM approaches are not directly applicable to these emerging technologies, the necessity of finding new algorithms and methods within the neuromorphic computing paradigm is the main challenge faced by researchers. Some promising approaches to applying event cameras to the SLAM problem have been identified in this paper, but future research focus needs to be applied to the problem of utilizing emerging neuromorphic processing capabilities to implement these methods practically and efficiently.

5.3. Neuromorphic SLAM Challenges

Developing SLAM algorithms that effectively utilize event-based data from event cameras and harness the computational capabilities of neuromorphic processors presents a significant challenge. These algorithms must be either heavily modified or newly conceived to fully exploit the strengths of both technologies. Furthermore, integrating data from event cameras with neuromorphic processors and other sensor modalities, such as IMUs or traditional cameras, necessitates the development of new fusion techniques. Managing the diverse data formats, temporal characteristics, and noise profiles from these sensors while maintaining consistency and accuracy throughout the SLAM process will be a complex task.

In terms of scalability, expanding event cameras and neuromorphic processor-based SLAM systems to accommodate large-scale environments with intricate dynamics will pose challenges in computational resource allocation. It is essential to ensure scalability while preserving real-time performance for practical deployment. Additionally, event cameras and neuromorphic processors must adapt to dynamic environments where scene changes occur rapidly. Developing algorithms capable of swiftly updating SLAM estimates based on incoming event data while maintaining robustness and accuracy is critical.

Leveraging the learning capabilities of neuromorphic processors for SLAM tasks, such as map building and localization, necessitates the design of training algorithms and methodologies proficient in learning from event data streams. The development of adaptive learning algorithms capable of enhancing SLAM performance over time in real-world environments presents a significant challenge. Moreover, ensuring the correctness and reliability of event camera and neuromorphic processor-based SLAM systems poses hurdles in verification and validation. Rigorous testing methodologies

must also be developed to validate the performance and robustness of these systems. If these challenges can be overcome, the potential rewards are significant, however.

Author Contributions: Conceptualization, S.T., A.R. and D.C.; methodology, S.T.; validation, S.T., A.R. and D.C.; formal analysis, S.T.; investigation, S.T.; writing—original draft preparation, S.T.; writing—review and editing, S.T., A.R. and D.C.; supervision, A.R. and D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This publication is partially supported by an ECU, School of Engineering Scholarship.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, X., et al., *Improving RGB-D SLAM accuracy in dynamic environments based on semantic and geometric constraints*. Measurement: Journal of the International Measurement Confederation, 2023. **217**.
2. Khan, M.S.A., et al., *Investigation of Widely Used SLAM Sensors Using Analytical Hierarchy Process*, in *Journal of Sensors*. 2022, Hindawi Limited.
3. Gelen, A.G. and A. Atasoy, *An Artificial Neural SLAM Framework for Event-Based Vision*. IEEE Access, 2023. **11**: p. 58436-58450.
4. Taheri, H. and Z.C. Xia, *SLAM; definition and evolution*. Engineering Applications of Artificial Intelligence, 2021. **97**: p. 104032.
5. Xia, Y., et al. *SLAM Back-End Optimization Algorithm Based on Vision Fusion IPS*. Sensors, 2022. **22**, DOI: 10.3390/s22239362.
6. Theodorou, C., et al., *Visual SLAM algorithms and their application for AR, mapping, localization and wayfinding*. Array, 2022. **15**.
7. Liu, L. and J.M. Aitken, *HFNet-SLAM: An Accurate and Real-Time Monocular SLAM System with Deep Features*. Sensors, 2023. **23**(4).
8. Gao, X. and T. Zhang, *Introduction to Visual SLAM From Theory to Practice Introduction to Visual SLAM Introduction to Visual SLAM*.
9. Gallego, G., et al., *Event-Based Vision: A Survey*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022. **44**(1): p. 154-180.
10. Chen, W., et al. *An Overview on Visual SLAM: From Tradition to Semantic*. Remote Sensing, 2022. **14**, DOI: 10.3390/rs14133010.
11. Aitsam, M., S. Davies, and A. Di Nuovo, *Neuromorphic Computing for Interactive Robotics: A Systematic Review*, in *IEEE Access*. 2022, Institute of Electrical and Electronics Engineers Inc. p. 122261-122279.
12. Cuadrado, J., et al., *Optical flow estimation from event-based cameras and spiking neural networks*. Frontiers in Neuroscience, 2023. **17**.
13. Fischer, T. and M. Milford, *Event-based visual place recognition with ensembles of temporal windows*. IEEE Robotics and Automation Letters, 2020. **5**(4): p. 6924-6931.
14. Furmonas, J., J. Liobe, and V. Barzdenas, *Analytical Review of Event-Based Camera Depth Estimation Methods and Systems*, in *Sensors*. 2022, MDPI.
15. Zhou, Y., G. Gallego, and S. Shen, *Event-Based Stereo Visual Odometry*. IEEE Transactions on Robotics, 2021. **37**(5): p. 1433-1450.
16. Ghosh, S. and G. Gallego, *Multi-Event-Camera Depth Estimation and Outlier Rejection by Refocused Events Fusion*. Advanced Intelligent Systems, 2022. **4**(12): p. 2200221-2200221.
17. Liu, Z., et al., *T-ESVO: Improved Event-Based Stereo Visual Odometry via Adaptive Time-Surface and Truncated Signed Distance Function*. Advanced Intelligent Systems, 2023.
18. Chen, P., W. Guan, and P. Lu, *ESVIO: Event-Based Stereo Visual Inertial Odometry*. IEEE Robotics and Automation Letters, 2023. **8**(6): p. 3661-3668.
19. Huang, K., et al., *Event-based Simultaneous Localization and Mapping: A Comprehensive Survey*. Computer Vision and Pattern Recognition, 2023. **14**(8).
20. Chen, G., et al., *Event-Based Neuromorphic Vision for Autonomous Driving: A Paradigm Shift for Bio-Inspired Visual Sensing and Perception*. IEEE Signal Processing Magazine, 2020. **37**(4): p. 34-49.
21. Fischer, T. and M. Milford, *How Many Events Do You Need? Event-Based Visual Place Recognition Using Sparse But Varying Pixels*. IEEE Robotics and Automation Letters, 2022. **7**(4): p. 12275-12282.
22. Favorskaya, M.N., *Deep Learning for Visual SLAM: The State-of-the-Art and Future Trends*, in *Electronics (Switzerland)*. 2023, MDPI.

23. Amir, A., et al. *A Low Power, Fully Event-Based Gesture Recognition System*. in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.
24. Yu, F., et al., *NeuroSLAM: a brain-inspired SLAM system for 3D environments*. Biological Cybernetics, 2019. **113**(5-6): p. 515-545.
25. Yang, Y., et al., *Neuromorphic electronics for robotic perception, navigation and control: A survey*. Engineering Applications of Artificial Intelligence, 2023. **126**: p. 106838-106838.
26. Schuman, C.D., et al., *Opportunities for neuromorphic computing algorithms and applications*, in *Nature Computational Science*. 2022, Springer Nature. p. 10-19.
27. Renner, A., et al., *Neuromorphic Visual Odometry with Resonator Networks*. Robotics, 2022.
28. Davies, M., et al., *Advancing Neuromorphic Computing With Loihi: A Survey of Results and Outlook*. Proceedings of the IEEE, 2021. **109**(5): p. 911-934.
29. Nunes, J.D., et al., *Spiking Neural Networks: A Survey*. IEEE Access, 2022. **10**: p. 60738-60764.
30. Hewawasam, H.S., M.Y. Ibrahim, and G.K. Appuhamillage, *Past, Present and Future of Path-Planning Algorithms for Mobile Robot Navigation in Dynamic Environments*. IEEE Open Journal of the Industrial Electronics Society, 2022. **3**: p. 353-365.
31. Furmonas, J., J. Liobe, and V. Barzdenas *Analytical Review of Event-Based Camera Depth Estimation Methods and Systems*. Sensors, 2022. **22**, DOI: 10.3390/s22031201.
32. Schuman, C.D., et al., *A Survey of Neuromorphic Computing and Neural Networks in Hardware*. Neural and Evolutionary Computing, 2017.
33. Pham, M.D., et al., *From Brain Models to Robotic Embodied Cognition: How Does Biological Plausibility Inform Neuromorphic Systems?* Brain Sciences, 2023. **13**(9): p. 1316-1316.
34. Saleem, H., R. Malekian, and H. Munir, *Neural Network-Based Recent Research Developments in SLAM for Autonomous Ground Vehicles: A Review*. IEEE Sensors Journal, 2023.
35. Macario Barros, A., et al., *A Comprehensive Survey of Visual SLAM Algorithms*, in *Robotics*. 2022, MDPI.
36. Zhang, Y., et al., *Review of Visual Simultaneous Localization and Mapping Based on Deep Learning*, in *Remote Sensing*. 2023, MDPI.
37. Tourani, A., et al. *Visual SLAM: What Are the Current Trends and What to Expect?* Sensors, 2022. **22**, DOI: 10.3390/s22239297.
38. Bavle, H., et al., *From SLAM to Situational Awareness: Challenges and Survey*, in *Sensors*. 2023, MDPI.
39. Dumont, N.S.Y., et al., *Exploiting semantic information in a spiking neural SLAM system*. Frontiers in Neuroscience, 2023. **17**.
40. Davison, A.J., et al., *MonoSLAM: Real-Time Single Camera SLAM*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007. **29**(6): p. 1052-1067.
41. Klein, G. and D. Murray. *Parallel Tracking and Mapping for Small AR Workspaces*. in 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality. 2007.
42. Newcombe, R.A., S.J. Lovegrove, and A.J. Davison. *DTAM: Dense tracking and mapping in real-time*. in 2011 International Conference on Computer Vision. 2011.
43. Salas-Moreno, R.F., et al. *SLAM++: Simultaneous Localisation and Mapping at the Level of Objects*. in 2013 IEEE Conference on Computer Vision and Pattern Recognition. 2013.
44. Ai, Y., et al., *DDL-SLAM: A Robust RGB-D SLAM in Dynamic Environments Combined With Deep Learning*. IEEE Access, 2020. **8**: p. 162335-162342.
45. Li, B., et al., *TextSLAM: Visual SLAM With Semantic Planar Text Features*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024. **46**(1): p. 593-610.
46. Liao, Z., et al., *SO-SLAM: Semantic Object SLAM With Scale Proportional and Symmetrical Texture Constraints*. IEEE Robotics and Automation Letters, 2022. **7**(2): p. 4008-4015.
47. Yang, C., et al., *SDF-SLAM: A Deep Learning Based Highly Accurate SLAM Using Monocular Camera Aiming at Indoor Map Reconstruction With Semantic and Depth Fusion*. IEEE Access, 2022. **10**: p. 10259-10272.
48. Lim, H., J. Jeon, and H. Myung, *UV-SLAM: Unconstrained Line-Based SLAM Using Vanishing Points for Structural Mapping*. IEEE Robotics and Automation Letters, 2022. **7**(2): p. 1518-1525.
49. Ran, T., et al., *RS-SLAM: A Robust Semantic SLAM in Dynamic Environments Based on RGB-D Sensor*. IEEE Sensors Journal, 2021. **21**(18): p. 20657-20664.
50. Liu, Y. and J. Miura, *RDMO-SLAM: Real-Time Visual SLAM for Dynamic Environments Using Semantic Label Prediction With Optical Flow*. IEEE Access, 2021. **9**: p. 106981-106997.
51. Liu, Y. and J. Miura, *RDS-SLAM: Real-Time Dynamic SLAM Using Semantic Segmentation Methods*. IEEE Access, 2021. **9**: p. 23772-23785.
52. Campos, C., et al., *ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM*. IEEE Transactions on Robotics, 2021. **37**(6): p. 1874-1890.
53. Li, Y., et al., *Structure-SLAM: Low-Drift Monocular SLAM in Indoor Environments*. IEEE Robotics and Automation Letters, 2020. **5**(4): p. 6583-6590.
54. Bavle, H., et al., *VPS-SLAM: Visual Planar Semantic SLAM for Aerial Robotic Systems*. IEEE Access, 2020. **8**: p. 60704-60718.

55. Gomez-Ojeda, R., et al., *PL-SLAM: A Stereo SLAM System Through the Combination of Points and Line Segments*. IEEE Transactions on Robotics, 2019. **35**(3): p. 734-746.
56. Mur-Artal, R. and J.D. Tardós, *ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras*. IEEE Transactions on Robotics, 2017. **33**(5): p. 1255-1262.
57. Mur-Artal, R., J.M.M. Montiel, and J.D. Tardos, *ORB-SLAM: A Versatile and Accurate Monocular SLAM System*. IEEE Transactions on Robotics, 2015. **31**(5): p. 1147-1163.
58. Engel, J., T. Schöps, and D. Cremers. *LSD-SLAM: Large-Scale Direct Monocular SLAM*. in *Computer Vision – ECCV 2014*. 2014. Cham: Springer International Publishing.
59. Boahen, K.A., *A burst-mode word-serial address-event link-I: transmitter design*. IEEE Transactions on Circuits and Systems I: Regular Papers, 2004. **51**(7): p. 1269-1280.
60. Kim, H., S. Leutenegger, and A.J. Davison. *Real-Time 3D Reconstruction and 6-DoF Tracking with an Event Camera*. in *Computer Vision – ECCV 2016*. 2016. Cham: Springer International Publishing.
61. Gehrig, M., et al. *Event-Based Angular Velocity Regression with Spiking Networks*. in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020.
62. Censi, A. and D. Scaramuzza. *Low-latency event-based visual odometry*. in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 2014.
63. Alzugaray, I. and M. Chli. *Asynchronous Multi-Hypothesis Tracking of Features with Event Cameras*. in *2019 International Conference on 3D Vision (3DV)*. 2019.
64. Kueng, B., et al. *Low-latency visual odometry using event-based feature tracks*. in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016.
65. Ye, C., et al. *Unsupervised Learning of Dense Optical Flow, Depth and Egomotion with Event-Based Sensors*. in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020.
66. Chaney, K., A.Z. Zhu, and K. Daniilidis. *Learning Event-Based Height From Plane and Parallax*. in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019.
67. Nguyen, A., et al. *Real-Time 6DOF Pose Relocalization for Event Cameras With Stacked Spatial LSTM Networks*. in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2019.
68. Hidalgo-Carrió, J., D. Gehrig, and D. Scaramuzza. *Learning Monocular Dense Depth from Events*. in *2020 International Conference on 3D Vision (3DV)*. 2020.
69. Gehrig, D., et al., *Combining Events and Frames Using Recurrent Asynchronous Multimodal Networks for Monocular Depth Prediction*. IEEE Robotics and Automation Letters, 2021. **6**(2): p. 2822-2829.
70. Zhu, A.Z., et al., *The Multivehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception*. IEEE Robotics and Automation Letters, 2018. **3**(3): p. 2032-2039.
71. Li, Z. and N. Snavely. *MegaDepth: Learning Single-View Depth Prediction from Internet Photos*. in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018.
72. Clady, X., S.-H. Ieng, and R. Benosman, *Asynchronous event-based corner detection and matching*. Neural Networks, 2015. **66**: p. 91-106.
73. Clady, X., et al., *A motion-based feature for event-based pattern recognition*. Frontiers in Neuroscience, 2017. **10**(JAN).
74. Vasco, V., A. Glover, and C. Bartolozzi. *Fast event-based Harris corner detection exploiting the advantages of event-driven cameras*. in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016.
75. Scheerlinck, C., N. Barnes, and R. Mahony, *Asynchronous Spatial Image Convolutions for Event Cameras*. IEEE Robotics and Automation Letters, 2019. **4**(2): p. 816-822.
76. Mueggler, E., C. Bartolozzi, and D. Scaramuzza. *Fast Event-based Corner Detection*. in *British Machine Vision Conference*. 2017.
77. Zhu, A.Z., N. Atanasov, and K. Daniilidis. *Event-based feature tracking with probabilistic data association*. in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017.
78. Zhu, A.Z., N. Atanasov, and K. Daniilidis. *Event-Based Visual Inertial Odometry*. in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.
79. Vidal, A.R., et al., *Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High-Speed Scenarios*. IEEE Robotics and Automation Letters, 2018. **3**(2): p. 994-1001.
80. Manderscheid, J., et al. *Speed Invariant Time Surface for Learning to Detect Corner Points With Event-Based Cameras*. in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.
81. Chibberre, P., et al. *Detecting Stable Keypoints from Events through Image Gradient Prediction*. in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2021.
82. Gioi, R.G.v., et al., *LSD: A Fast Line Segment Detector with a False Detection Control*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010. **32**(4): p. 722-732.
83. Everding, L. and J. Conradt, *Low-latency line tracking using event-based dynamic vision sensors*. Frontiers in Neuroinformatics, 2018. **12**(FEB).
84. Wenzhen, Y. and S. Ramalingam. *Fast localization and tracking using event sensors*. in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016.

85. Bertrand, J., A. Yiğit, and S. Durand. *Embedded Event-based Visual Odometry*. in 2020 6th International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP). 2020.
86. Chamorro, W., J. Solà, and J. Andrade-Cetto, *Event-Based Line SLAM in Real-Time*. IEEE Robotics and Automation Letters, 2022. 7(3): p. 8146-8153.
87. Guan, W., et al., *PL-EVIO: Robust Monocular Event-Based Visual Inertial Odometry With Point and Line Features*. IEEE Transactions on Automation Science and Engineering, 2023: p. 1-17.
88. Brändli, C., et al. *ELiSeD - An event-based line segment detector*. in 2016 Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP). 2016.
89. Valeiras, D.R., et al., *Event-Based Line Fitting and Segment Detection Using a Neuromorphic Visual Sensor*. IEEE Transactions on Neural Networks and Learning Systems, 2019. 30(4): p. 1218-1230.
90. Besl, P.J. and N.D. McKay, *A method for registration of 3-D shapes*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992. 14(2): p. 239-256.
91. Gehrig, D., et al., *EKLT: Asynchronous Photometric Feature Tracking Using Events and Frames*. International Journal of Computer Vision, 2020. 128(3): p. 601-618.
92. Seok, H. and J. Lim. *Robust Feature Tracking in DVS Event Stream using Bézier Mapping*. in 2020 IEEE Winter Conference on Applications of Computer Vision (WACV). 2020.
93. Chui, J., S. Klenk, and D. Cremers, *Event-Based Feature Tracking in Continuous Time with Sliding Window Optimization*. Computer Vision and Pattern Recognition, 2021.
94. Alzugaray, I. and M. Chli. *HASTE: multi-Hypothesis Asynchronous Speeded-up Tracking of Events*. in 31st British Machine Vision Virtual Conference (BMVC 2020). 2020. ETH Zurich, Institute of Robotics and Intelligent Systems.
95. Hadviger, A., et al. *Feature-based Event Stereo Visual Odometry*. in 2021 European Conference on Mobile Robots (ECMR). 2021.
96. Hu, S., et al. *eCDT: Event Clustering for Simultaneous Feature Detection and Tracking*. in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2022.
97. Messikommer, N., et al. *Data-Driven Feature Tracking for Event Cameras*. in 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2023.
98. Hidalgo-Carrió, J., G. Gallego, and D. Scaramuzza. *Event-aided Direct Sparse Odometry*. in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022.
99. Rebecq, H., et al., *EVO: A Geometric Approach to Event-Based 6-DOF Parallel Tracking and Mapping in Real Time*. IEEE Robotics and Automation Letters, 2017. 2(2): p. 593-600.
100. Rebecq, H., et al., *EMVS: Event-Based Multi-View Stereo—3D Reconstruction with an Event Camera in Real-Time*. International Journal of Computer Vision, 2018. 126(12): p. 1394-1414.
101. Zuo, Y.F., et al., *DEVO: Depth-Event Camera Visual Odometry in Challenging Conditions*, in 2022 International Conference on Robotics and Automation (ICRA). 2022, IEEE Press: Philadelphia, PA, USA. p. 2179–2185.
102. Gallego, G., H. Rebecq, and D. 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. 2018.
103. Gu, C., et al. *The Spatio-Temporal Poisson Point Process: A Simple Model for the Alignment of Event Camera Data*. in 2021 IEEE/CVF International Conference on Computer Vision (ICCV). 2021.
104. Zhou, T., et al. *Unsupervised Learning of Depth and Ego-Motion from Video*. in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.
105. Wang, S., et al. *DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks*. in 2017 IEEE International Conference on Robotics and Automation (ICRA). 2017.
106. Yang, N., et al. *D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry*. in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020.
107. Zhang, S., J. Zhang, and D. Tao. *Towards Scale Consistent Monocular Visual Odometry by Learning from the Virtual World*. in 2022 International Conference on Robotics and Automation (ICRA). 2022.
108. Zhao, H., et al. *JPerceiver: Joint Perception Network for Depth, Pose and Layout Estimation in Driving Scenes*. in Computer Vision – ECCV 2022. 2022. Cham: Springer Nature Switzerland.
109. Zhang, S., J. Zhang, and D. Tao. *Towards Scale-Aware, Robust, and Generalizable Unsupervised Monocular Depth Estimation by Integrating IMU Motion Dynamics*. in Computer Vision – ECCV 2022. 2022. Cham: Springer Nature Switzerland.
110. Zhu, A.Z., et al. *Unsupervised Event-Based Learning of Optical Flow, Depth, and Egomotion*. in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019.
111. He, W., et al., *Comparing SNNs and RNNs on neuromorphic vision datasets: Similarities and differences*. Neural Networks, 2020. 132: p. 108-120.
112. Deng, L., et al., *Rethinking the performance comparison between SNNs and ANNs*. Neural Networks, 2020. 121: p. 294-307.
113. Iyer, L.R., Y. Chua, and H. Li, *Is Neuromorphic MNIST neuromorphic? Analyzing the discriminative power of neuromorphic datasets in the time domain*. 2018.

114. Kim, Y. and P. Panda, *Optimizing Deeper Spiking Neural Networks for Dynamic Vision Sensing*. Neural Networks, 2021. **144**: p. 686-698.
115. Yao, M., et al., *Sparser spiking activity can be better: Feature Refine-and-Mask spiking neural network for event-based visual recognition*. Neural Networks, 2023. **166**: p. 410-423.
116. Mueggler, E., et al., *The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM*. The International Journal of Robotics Research, 2017. **36**(2): p. 142-149.
117. Zhou, Y., et al. *Semi-dense 3D Reconstruction with a Stereo Event Camera*. in *Computer Vision – ECCV 2018*. 2018. Cham: Springer International Publishing.
118. Guan, W., et al., *PL-EVIO: Robust Monocular Event-based Visual Inertial Odometry with Point and Line Features*.
119. Delmerico, J., et al. *Are We Ready for Autonomous Drone Racing? The UZH-FPV Drone Racing Dataset*. in *2019 International Conference on Robotics and Automation (ICRA)*. 2019.
120. Klenk, S., et al. *TUM-VIE: The TUM Stereo Visual-Inertial Event Dataset*. in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021.
121. Gao, L., et al., *VECTOR: A Versatile Event-Centric Benchmark for Multi-Sensor SLAM*. IEEE Robotics and Automation Letters, 2022. **7**(3): p. 8217-8224.
122. Yin, J., et al., *M2DGR: A Multi-Sensor and Multi-Scenario SLAM Dataset for Ground Robots*. IEEE Robotics and Automation Letters, 2022. **7**(2): p. 2266-2273.
123. Sturm, J., et al. *A benchmark for the evaluation of RGB-D SLAM systems*. in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012.
124. Gallego, G., et al., *Event-Based, 6-DOF Camera Tracking from Photometric Depth Maps*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018. **40**(10): p. 2402-2412.
125. Nunes, U.M. and Y. Demiris. *Entropy Minimisation Framework for Event-Based Vision Model Estimation*. in *Computer Vision – ECCV 2020*. 2020. Cham: Springer International Publishing.
126. Nunes, U.M. and Y. Demiris, *Robust Event-Based Vision Model Estimation by Dispersion Minimisation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022. **44**(12): p. 9561-9573.
127. Bryner, S., et al. *Event-based, Direct Camera Tracking from a Photometric 3D Map using Nonlinear Optimization*. in *2019 International Conference on Robotics and Automation (ICRA)*. 2019.
128. Engel, J., V. Koltun, and D. Cremers, *Direct Sparse Odometry*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018. **40**(3): p. 611-625.
129. Rebecq, H., et al., *High Speed and High Dynamic Range Video with an Event Camera*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021. **43**(6): p. 1964-1980.
130. Guan, W. and P. Lu. *Monocular Event Visual Inertial Odometry based on Event-corner using Sliding Windows Graph-based Optimization*. in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022.
131. LeCun, Y., Y. Bengio, and G. Hinton, *Deep learning*. Nature, 2015. **521**(7553): p. 436-444.
132. Falotico, E., et al., *Connecting Artificial Brains to Robots in a Comprehensive Simulation Framework: The NeuroRobotics Platform*. Frontiers in NeuroRobotics, 2017. **11**.
133. Roy, K., A. Jaiswal, and P. Panda, *Towards spike-based machine intelligence with neuromorphic computing*. Nature, 2019. **575**(7784): p. 607-617.
134. Furber, S., *Large-scale neuromorphic computing systems*. Journal of Neural Engineering, 2016. **13**(5): p. 051001.
135. Nandakumar, S.R., et al., *Building Brain-Inspired Computing Systems: Examining the Role of Nanoscale Devices*. IEEE Nanotechnology Magazine, 2018. **12**(3): p. 19-35.
136. Bartolozzi, C., G. Indiveri, and E. Donati, *Embodied neuromorphic intelligence*. Nature Communications, 2022. **13**(1): p. 1024.
137. Davies, M., et al., *Loihi: A Neuromorphic Manycore Processor with On-Chip Learning*. IEEE Micro, 2018. **38**(1): p. 82-99.
138. James, C.D., et al., *A historical survey of algorithms and hardware architectures for neural-inspired and neuromorphic computing applications*, in *Biologically Inspired Cognitive Architectures*. 2017, Elsevier B.V. p. 49-64.
139. Strukov, D., et al., *Building brain-inspired computing*. Nature Communications, 2019. **10**(1): p. 4838.
140. Thakur, C.S., et al., *Large-Scale Neuromorphic Spiking Array Processors: A Quest to Mimic the Brain*, in *Frontiers in Neuroscience*. 2018, Frontiers Media S.A.
141. Furber, S.B., et al., *The SpiNNaker Project*. Proceedings of the IEEE, 2014. **102**(5): p. 652-665.
142. Furber, S., *Large-scale neuromorphic computing systems*, in *Journal of Neural Engineering*. 2016, Institute of Physics Publishing.
143. Schemmel, J., et al. *A wafer-scale neuromorphic hardware system for large-scale neural modeling*. in *2010 IEEE International Symposium on Circuits and Systems (ISCAS)*. 2010.
144. Schmitt, S., et al. *Neuromorphic hardware in the loop: Training a deep spiking network on the BrainScaleS wafer-scale system*. in *2017 International Joint Conference on Neural Networks (IJCNN)*. 2017.
145. DeBole, M.V., et al., *TrueNorth: Accelerating From Zero to 64 Million Neurons in 10 Years*. Computer, 2019. **52**(5): p. 20-29.

146. *Brainchip*. 2023 [cited 2023 4/10/2023]; Available from: <https://brainchip.com/akida-generations/>.
147. Vanarse, A., A. Osseiran, and A. Rassau, *Neuromorphic engineering — A paradigm shift for future IM technologies*. IEEE Instrumentation & Measurement Magazine, 2019. **22**(2): p. 4-9.
148. Höppner, S., et al., *The SpiNNaker 2 Processing Element Architecture for Hybrid Digital Neuromorphic Computing*. Hardware Architecture, 2021.
149. Ivanov, D., et al., *Neuromorphic artificial intelligence systems*. Frontiers in Neuroscience, 2022. **16**.
150. Mayr, C., S. Hoeppe, and S.B. Furber, *SpiNNaker 2: A 10 Million Core Processor System for Brain Simulation and Machine Learning*. Emerging Technologies, 2019.
151. van Albada, S.J., et al., *Performance Comparison of the Digital Neuromorphic Hardware SpiNNaker and the Neural Network Simulation Software NEST for a Full-Scale Cortical Microcircuit Model*. Frontiers in Neuroscience, 2018. **12**.
152. Merolla, P.A., et al., *Artificial brains. A million spiking-neuron integrated circuit with a scalable communication network and interface*. Science, 2014. **345**(6197): p. 668-73.
153. Andreopoulos, A., et al. *A Low Power, High Throughput, Fully Event-Based Stereo System*. in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018.
154. DeWolf, T., P. Jaworski, and C. Eliasmith, *Nengo and Low-Power AI Hardware for Robust, Embedded Neurorobotics*. Frontiers in Neurorobotics, 2020. **14**.
155. Stagsted, R.K., et al. *Event-based PID controller fully realized in neuromorphic hardware: A one DoF study*. in IEEE International Conference on Intelligent Robots and Systems. 2020. Institute of Electrical and Electronics Engineers Inc.
156. *Lava software framework*. [cited 2024 30/4]; Available from: <https://lava-nc.org/>.
157. Gröbl, A., et al., *Verification and Design Methods for the BrainScaleS Neuromorphic Hardware System*. Journal of Signal Processing Systems, 2020. **92**(11): p. 1277-1292.
158. Wunderlich, T., et al., *Demonstrating Advantages of Neuromorphic Computation: A Pilot Study*. Frontiers in Neuroscience, 2019. **13**.
159. Merolla, P., et al. *A digital neurosynaptic core using embedded crossbar memory with 45pJ per spike in 45nm*. in 2011 IEEE Custom Integrated Circuits Conference (CICC). 2011.
160. Stomatias, E., et al. *Scalable energy-efficient, low-latency implementations of trained spiking Deep Belief Networks on SpiNNaker*. in 2015 International Joint Conference on Neural Networks (IJCNN). 2015.
161. Querlioz, D., O. Bichler, and C. Gamrat. *Simulation of a memristor-based spiking neural network immune to device variations*. in The 2011 International Joint Conference on Neural Networks. 2011.
162. Cramer, B., et al., *Surrogate gradients for analog neuromorphic computing*. Proceedings of the National Academy of Sciences, 2022. **119**(4): p. e2109194119.
163. Schreiber, K., et al., *Closed-loop experiments on the BrainScaleS-2 architecture*, in Proceedings of the 2020 Annual Neuro-Inspired Computational Elements Workshop. 2020, Association for Computing Machinery: Heidelberg, Germany. p. Article 17.
164. Moradi, S., et al., *A Scalable Multicore Architecture With Heterogeneous Memory Structures for Dynamic Neuromorphic Asynchronous Processors (DYNAPs)*. IEEE Transactions on Biomedical Circuits and Systems, 2017. **12**: p. 106-122.
165. Vanarse, A., et al. *A Hardware-Deployable Neuromorphic Solution for Encoding and Classification of Electronic Nose Data*. Sensors, 2019. **19**, DOI: 10.3390/s19224831.
166. Wang, X., X. Lin, and X. Dang, *Supervised learning in spiking neural networks: A review of algorithms and evaluations*. Neural Networks, 2020. **125**: p. 258-280.
167. McCulloch, W.S. and W. Pitts, *A logical calculus of the ideas immanent in nervous activity*. The bulletin of mathematical biophysics, 1943. **5**(4): p. 115-133.
168. Guo, W., et al., *Neural Coding in Spiking Neural Networks: A Comparative Study for Robust Neuromorphic Systems*. Frontiers in Neuroscience, 2021. **15**.
169. Zhang, T., et al., *Tuning Convolutional Spiking Neural Network With Biologically Plausible Reward Propagation*. IEEE Transactions on Neural Networks and Learning Systems, 2022. **33**(12): p. 7621-7631.
170. Cordone, L., *Performance of spiking neural networks on event data for embedded automotive applications*. 2022, Université Côte d'Azur.
171. Yang, L., et al., *Coreset: Hierarchical neuromorphic computing supporting large-scale neural networks with improved resource efficiency*. Neurocomputing, 2022. **474**: p. 128-140.
172. Tang, G., A. Shah, and K.P. Michmizos. *Spiking Neural Network on Neuromorphic Hardware for Energy-Efficient Unidimensional SLAM*. in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2019.
173. Shrestha, A., et al., *A Survey on Neuromorphic Computing: Models and Hardware*. IEEE Circuits and Systems Magazine, 2022. **22**(2): p. 6-35.
174. Kreiser, R., et al. *A Neuromorphic Approach to Path Integration: A Head-Direction Spiking Neural Network with Vision-driven Reset*. in 2018 IEEE International Symposium on Circuits and Systems (ISCAS). 2018.

175. Kreiser, R., et al. *Pose Estimation and Map Formation with Spiking Neural Networks: towards Neuromorphic SLAM*. in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2018.
176. Kreiser, R., et al. *Error estimation and correction in a spiking neural network for map formation in neuromorphic hardware*. in 2020 IEEE International Conference on Robotics and Automation (ICRA). 2020.
177. Aboumerhi, K., et al., *Neuromorphic applications in medicine*. Journal of Neural Engineering, 2023. **20**(4): p. 041004.
178. Gordon, C., et al. *An artificial synapse for interfacing to biological neurons*. in 2006 IEEE International Symposium on Circuits and Systems. 2006.
179. Boi, F., et al., *A Bidirectional Brain-Machine Interface Featuring a Neuromorphic Hardware Decoder*. Frontiers in Neuroscience, 2016. **10**.
180. Nurse, E., et al., *Decoding EEG and LFP signals using deep learning: heading TrueNorth*, in *Proceedings of the ACM International Conference on Computing Frontiers*. 2016, Association for Computing Machinery: Como, Italy. p. 259–266.
181. Jung, R., E.J. Brauer, and J.J. Abbas, *Real-time interaction between a neuromorphic electronic circuit and the spinal cord*. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2001. **9**(3): p. 319-326.
182. Corradi, F., D. Bontrager, and G. Indiveri. *Toward neuromorphic intelligent brain-machine interfaces: An event-based neural recording and processing system*. in 2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings. 2014.
183. Corradi, F. and G. Indiveri, *A Neuromorphic Event-Based Neural Recording System for Smart Brain-Machine-Interfaces*. IEEE Transactions on Biomedical Circuits and Systems, 2015. **9**(5): p. 699-709.
184. Sinha, S., et al. *A workload-aware neuromorphic controller for dynamic power and thermal management*. in 2011 NASA/ESA Conference on Adaptive Hardware and Systems (AHS). 2011.
185. Jin, L. and M. Brooke. *A fully parallel learning neural network chip for real-time control*. in *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339)*. 1999.
186. Liu, J. and M. Brooke. *Fully parallel on-chip learning hardware neural network for real-time control*. in 1999 IEEE International Symposium on Circuits and Systems (ISCAS). 1999.
187. Dong, Z., et al., *A novel memristive multilayer feedforward small-world neural network with its applications in PID control*. ScientificWorldJournal, 2014. **2014**: p. 394828.
188. Rocke, P., et al. *Investigating the Suitability of FPAAs for Evolved Hardware Spiking Neural Networks*. in *Evolvable Systems: From Biology to Hardware*. 2008. Berlin, Heidelberg: Springer Berlin Heidelberg.
189. Dean, M.E., et al. *An Application Development Platform for neuromorphic computing*. in 2016 International Joint Conference on Neural Networks (IJCNN). 2016.
190. Schuman, C.D., et al. *Parallel Evolutionary Optimization for Neuromorphic Network Training*. in 2016 2nd Workshop on Machine Learning in HPC Environments (MLHPC). 2016.
191. Galluppi, F., et al., *A hierarchical configuration system for a massively parallel neural hardware platform*, in *Proceedings of the 9th conference on Computing Frontiers*. 2012, Association for Computing Machinery: Cagliari, Italy. p. 183–192.
192. Arthur, J.V., et al. *Building block of a programmable neuromorphic substrate: A digital neurosynaptic core*. in *The 2012 International Joint Conference on Neural Networks (IJCNN)*. 2012.
193. Ahmed, K., et al. *System Design for In-Hardware STDP Learning and Spiking Based Probabilistic Inference*. in 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI). 2016.
194. Georgiou, J., A.G. Andreou, and P.O. Pouliquen. *A mixed analog/digital asynchronous processor for cortical computations in 3D SOI-CMOS*. in 2006 IEEE International Symposium on Circuits and Systems. 2006.
195. Shih-Chii, L. and M. Oster. *Feature competition in a spike-based winner-take-all VLSI network*. in 2006 IEEE International Symposium on Circuits and Systems (ISCAS). 2006.
196. Glackin, B., et al. *Emulating Spiking Neural Networks for edge detection on FPGA hardware*. in 2009 International Conference on Field Programmable Logic and Applications. 2009.
197. Goldberg, D.H., G. Cauwenberghs, and A.G. Andreou, *Probabilistic synaptic weighting in a reconfigurable network of VLSI integrate-and-fire neurons*. Neural Networks, 2001. **14**(6): p. 781-793.
198. Xiong, Y., et al. *An analog CMOS pulse coupled neural network for image segmentation*. in 2010 10th IEEE International Conference on Solid-State and Integrated Circuit Technology. 2010.
199. Secco, J., et al. *Memristor cellular automata through belief propagation inspired algorithm*. in 2015 International SoC Design Conference (ISOCC). 2015.
200. Bohrn, M., L. Fucik, and R. Vrba. *Field Programmable Neural Array for feed-forward neural networks*. in 2013 36th International Conference on Telecommunications and Signal Processing (TSP). 2013.
201. Fang, W.C., et al., *A VLSI neural processor for image data compression using self-organization networks*. IEEE Transactions on Neural Networks, 1992. **3**(3): p. 506-518.
202. Martínez, J.J., F.J. Toledo, and J.M. Ferrández. *New emulated discrete model of CNN architecture for FPGA and DSP applications*. in *Artificial Neural Nets Problem Solving Methods*. 2003. Berlin, Heidelberg: Springer Berlin Heidelberg.

203. Shi, B.E., et al. *Expandable hardware for computing cortical feature maps*. in 2006 IEEE International Symposium on Circuits and Systems (ISCAS). 2006.
204. Ielmini, D., et al. *Neuromorphic computing with hybrid memristive/CMOS synapses for real-time learning*. in 2016 IEEE International Symposium on Circuits and Systems (ISCAS). 2016.
205. Ranjan, R., et al. *Integrated circuit with memristor emulator array and neuron circuits for neuromorphic pattern recognition*. in 2016 39th International Conference on Telecommunications and Signal Processing (TSP). 2016.
206. Hu, M., et al. *Hardware realization of BSB recall function using memristor crossbar arrays*. in DAC Design Automation Conference 2012. 2012.
207. Tarkov, M.S. *Hopfield Network with Interneuron Connections Based on Memristor Bridges*. in *Advances in Neural Networks – ISNN 2016*. 2016. Cham: Springer International Publishing.
208. Soltiz, M., et al., *Memristor-Based Neural Logic Blocks for Nonlinearly Separable Functions*. IEEE Transactions on Computers, 2013. **62**(08): p. 1597-1606.
209. Zhang, P., et al., *Forgetting memristor based neuromorphic system for pattern training and recognition*. Neurocomputing, 2017. **222**: p. 47-53.
210. Darwish, M., et al., *Ultracompact Graphene Multigate Variable Resistor for Neuromorphic Computing*. IEEE Transactions on Nanotechnology, 2016. **15**(2): p. 318-327.
211. Wen, M.B.Q.W.W. and J. Rajendran, *Security of neuromorphic computing: Thwarting learning attacks using memristor's obsolescence effect*, in 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). 2016, IEEE Press: Chaofer Yang Beiyue Liu Hai Li Yiran Chen: Austin, TX, USA. p. 1-6.
212. Vitabile, S., et al. *Efficient MLP digital implementation on FPGA*. in 8th Euromicro Conference on Digital System Design (DSD'05). 2005.
213. Garbin, D., et al. *On the impact of OxRAM-based synapses variability on convolutional neural networks performance*. in *Proceedings of the 2015 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH'15)*. 2015.
214. Fieres, J., J. Schemmel, and K. Meier. *Training convolutional networks of threshold neurons suited for low-power hardware implementation*. in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. 2006.
215. Dhawan, K., S.P. R, and N. R. K, *Identification of traffic signs for advanced driving assistance systems in smart cities using deep learning*. Multimedia Tools and Applications, 2023. **82**(17): p. 26465-26480.
216. Ibrayev, T., et al. *A design of HTM spatial pooler for face recognition using memristor-CMOS hybrid circuits*. in 2016 IEEE International Symposium on Circuits and Systems (ISCAS). 2016.
217. Knag, P., L. Chester, and Z. Zhengya. *A 1.40mm² 141mW 898GOPS sparse neuromorphic processor in 40nm CMOS*. in 2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits). 2016.
218. Farabet, C., et al. *CNP: An FPGA-based processor for Convolutional Networks*. in 2009 International Conference on Field Programmable Logic and Applications. 2009.
219. Le, T.H., *Applying Artificial Neural Networks for Face Recognition*. Advances in Artificial Neural Systems, 2011. **2011**: p. 673016.
220. Teodoro, Á.-S., A.Á.-C. Jesús, and H.-C. Roberto. *Detection of facial emotions using neuromorphic computation*. in *Proc.SPIE*. 2022.
221. Manakitsa, N., et al. *A Review of Machine Learning and Deep Learning for Object Detection, Semantic Segmentation, and Human Action Recognition in Machine and Robotic Vision*. Technologies, 2024. **12**, DOI: 10.3390/technologies12020015.
222. Adhikari, S.P., et al., *Memristor Bridge Synapse-Based Neural Network and Its Learning*. IEEE Transactions on Neural Networks and Learning Systems, 2012. **23**(9): p. 1426-1435.
223. Yang, Z., et al. *High-Speed Object Recognition Based on a Neuromorphic System*. Electronics, 2022. **11**, DOI: 10.3390/electronics11244179.
224. Yang, B., *Research on Vehicle Detection and Recognition Technology Based on Artificial Intelligence*. Microprocessors and Microsystems, 2023: p. 104937.
225. Wan, H., et al., *A Novel Neural Network Model for Traffic Sign Detection and Recognition under Extreme Conditions*. Journal of Sensors, 2021. **2021**: p. 9984787.
226. Mukundan, A., et al. *Air Pollution Detection Using a Novel Snap-Shot Hyperspectral Imaging Technique*. Sensors, 2022. **22**, DOI: 10.3390/s22166231.
227. Kow, P.-Y., et al., *Real-time image-based air quality estimation by deep learning neural networks*. Journal of Environmental Management, 2022. **307**: p. 114560.
228. Onorato, M., et al. *Non-linear circuit effects on analog VLSI neural network implementations*. in *Proceedings of the Fourth International Conference on Microelectronics for Neural Networks and Fuzzy Systems*. 1994.
229. Yang, J., et al. *Using Deep Learning to Detect Defects in Manufacturing: A Comprehensive Survey and Current Challenges*. Materials, 2020. **13**, DOI: 10.3390/ma13245755.
230. Aibe, N., et al., *Performance evaluation system for probabilistic neural network hardware*. Artificial Life and Robotics, 2004. **8**(2): p. 208-213.
231. Ceolini, E., et al., *Hand-Gesture Recognition Based on EMG and Event-Based Camera Sensor Fusion: A Benchmark in Neuromorphic Computing*. Frontiers in Neuroscience, 2020. **14**.

232. Liu, X. and C. Aldrich *Deep Learning Approaches to Image Texture Analysis in Material Processing*. Metals, 2022. **12**, DOI: 10.3390/met12020355.
233. Dixit, U., et al., *Texture classification using convolutional neural network optimized with whale optimization algorithm*. SN Applied Sciences, 2019. **1**(6): p. 655.
234. Roska, T., et al. *An Associative Memory with oscillatory CNN arrays using spin torque oscillator cells and spin-wave interactions architecture and End-to-end Simulator*. in *2012 13th International Workshop on Cellular Nanoscale Networks and their Applications*. 2012.
235. Khosla, D., Y. Chen, and K. Kim, *A neuromorphic system for video object recognition*. Frontiers in Computational Neuroscience, 2014. **8**.
236. Al-Salih, A.A.M. and S.I. Ahson. *Object detection and features extraction in video frames using direct thresholding*. in *2009 International Multimedia, Signal Processing and Communication Technologies*. 2009.
237. Wang, X., et al., *Real-time and accurate object detection in compressed video by long short-term feature aggregation*. Computer Vision and Image Understanding, 2021. **206**: p. 103188.
238. Zhu, H., et al. *A Review of Video Object Detection: Datasets, Metrics and Methods*. Applied Sciences, 2020. **10**, DOI: 10.3390/app10217834.
239. Koh, T.C., et al., *Towards efficient video-based action recognition: context-aware memory attention network*. SN Applied Sciences, 2023. **5**(12): p. 330.
240. Sánchez-Caballero, A., D. Fuentes-Jiménez, and C. Losada-Gutiérrez, *Real-time human action recognition using raw depth video-based recurrent neural networks*. Multimedia Tools and Applications, 2023. **82**(11): p. 16213-16235.
241. Tsai, J.-K., et al. *Deep Learning-Based Real-Time Multiple-Person Action Recognition System*. Sensors, 2020. **20**, DOI: 10.3390/s20174758.
242. Kim, H., et al. *An Effective Motion-Tracking Scheme for Machine-Learning Applications in Noisy Videos*. Applied Sciences, 2023. **13**, DOI: 10.3390/app13053338.
243. Liu, L., B. Lin, and Y. Yang, *Moving scene object tracking method based on deep convolutional neural network*. Alexandria Engineering Journal, 2024. **86**: p. 592-602.
244. Skrzypkowiak, S.S. and V.K. Jain. *Video motion estimation using a neural network*. in *Proceedings of IEEE International Symposium on Circuits and Systems - ISCAS '94*. 1994.
245. Botella, G. and C. García, *Real-time motion estimation for image and video processing applications*. Journal of Real-Time Image Processing, 2016. **11**(4): p. 625-631.
246. Lee, J., et al. *BlockNet: A Deep Neural Network for Block-Based Motion Estimation Using Representative Matching*. Symmetry, 2020. **12**, DOI: 10.3390/sym12050840.
247. Yoon, J.H. and A. Raychowdhury, *NeuroSLAM: A 65-nm 7.25-to-8.79-TOPS/W Mixed-Signal Oscillator-Based SLAM Accelerator for Edge Robotics*. IEEE Journal of Solid-State Circuits, 2021. **56**(1): p. 66-78.
248. Jones, A., et al., *A neuromorphic SLAM architecture using gated-memristive synapses*. Neurocomputing, 2020. **381**: p. 89-104.
249. Lee, J. and J.H. Yoon. *A Neuromorphic SLAM Accelerator Supporting Multi-Agent Error Correction in Swarm Robotics*. in *Proceedings - International SoC Design Conference 2022, ISOCC 2022*. 2022. Institute of Electrical and Electronics Engineers Inc.
250. Yoon, J.H. and A. Raychowdhury. *31.1 A 65nm 8.79TOPS/W 23.82mW Mixed-Signal Oscillator-Based NeuroSLAM Accelerator for Applications in Edge Robotics*. in *2020 IEEE International Solid-State Circuits Conference - (ISSCC)*. 2020.
251. Shen, D., et al., *ORB-NeuroSLAM: A Brain-Inspired 3-D SLAM System Based on ORB Features*. IEEE Internet of Things Journal, 2024. **11**(7): p. 12408-12418.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.