POLITECNICO DI TORINO

Master's Degree in Computer Engineering

Master's Degree Thesis

AMOS: Adaptive Motion Segmentation using Spiking Neural Networks with Short-Term Synaptic Plasticity

Supervisors
Prof. Stefano Di Carlo
Prof. Alessandro Savino
PhD. Alessio Carpegna
MSc. Alessio Caviglia

Candidate
Anil Bayram Gogebakan

December 2025

Summary

Neuromorphic computing aims to emulate the computational efficiency and adaptability of biological neural systems. By integrating memory and processing within the same physical substrate, neuromorphic architectures avoid the data movement bottlenecks that limit traditional von Neumann systems. Spiking neural networks form the algorithmic foundation of this paradigm, since they operate through discrete spikes that naturally encode time, sparsity, and causal relationships. Event-based cameras complement this form of computation by producing asynchronous streams of events that directly correspond to changes in the visual scene rather than capturing full frames at fixed intervals. Their high temporal resolution, low latency, and sparse output make them well aligned with spike based processing. This synergy creates an opportunity for designing visual perception systems that operate efficiently in dynamic and resource constrained environments.

Despite these advantages, event-based data captured in automotive or mobile scenarios contain both object motion and ego-motion. Ego-motion often produces large amounts of background activity, which can obscure independently moving objects such as cars, pedestrians, bicycles, and motorcycles. Extracting meaningful motion patterns from these dense event streams remains challenging, especially when lighting, contrast, and motion conditions vary rapidly. Many existing approaches rely on reconstruction, supervised learning, or hand tuned temporal heuristics. These strategies either reintroduce frame based computation or limit generalizability. The aim of this thesis is to address these shortcomings by developing a biologically inspired method that isolates independent motion directly from the event stream using only local spiking and synaptic dynamics.

The proposed framework, named AMOS, which stands for Adaptive Motion Segmentation using Spiking Neural Networks, is based on the idea that short-term synaptic plasticity can serve as an adaptive filtering mechanism for event-based data. Short-term plasticity temporarily changes the strength of synapses according to their recent activity. In particular, the Tsodyks Markram model provides two complementary mechanisms. Short-term depression reduces synaptic efficacy under sustained stimulation, which suppresses repetitive background activity originating from ego-motion. Short-term facilitation increases synaptic efficacy when presynaptic activity occurs in rapid succession, which may emphasize consistent or salient motion. By embedding these mechanisms into spiking convolutional layers implemented in the NEST simulator, AMOS filters event streams in real time, enhancing transient motion patterns while attenuating static and redundant ones.

To evaluate the contribution of synaptic plasticity, four main configurations were designed and compared. The first configuration used direct one to one connections without spatial interaction, serving as a baseline. The second configuration applied static convolutional filters that captured spatial neighborhoods but lacked temporal adaptation. The third configuration used depressing synapses, enabling automatic suppression of persistent background activity. The final configuration attempted a hybrid design combining

a facilitating center region with a depressing surround, inspired by receptive field structures observed in biological vision. Across all models, different membrane time constants and synaptic parameters were explored to examine their influence on temporal selectivity, responsiveness to motion, and robustness to noise.

Experiments were conducted using two publicly available automotive event-based datasets. The first dataset, the Prophesee 1 Megapixel Automotive Detection Dataset, provides high resolution recordings of urban, suburban, and highway driving conditions with extensive annotation of vehicles and other road users. The second dataset, MVSEC, includes synchronized event data, grayscale images, inertial measurements, and LiDAR readings. For both datasets, evaluation was performed on timestamps where ground truth bounding boxes were available. After filtering through the SNN based system, event clusters were grouped using the DBSCAN algorithm to generate candidate detections. These detections were evaluated through precision, recall, the F1 score, and the mean Intersection over Union metric.

Across experiments, AMOS revealed distinct behaviors among the four filtering models. Static filters achieved balanced precision and recall but generated high false positive counts because they did not attenuate ego-motion. Depressing filters significantly reduced false positives by adaptively suppressing sustained background activity. Although recall was lower, these filters produced the most reliable detections and demonstrated strong robustness to variations in lighting and motion. Hybrid filters provided mixed results. While they were designed to combine facilitation and depression, facilitation often accumulated too slowly in real scenes, leading to incomplete or inconsistent responses. In the Prophesee dataset, static filters achieved reasonable F1 scores under strict IoU thresholds, whereas depressing filters performed better under relaxed IoU thresholds, where the focus is on motion isolation. In the MVSEC dataset, depressing synapses again outperformed all other configurations, achieving the highest F1 scores across different IoU thresholds.

These results underline an important scientific observation. The reduction of false positives, rather than absolute recall, appears to be the most meaningful measure of performance for this work. One limitation arises from the fact that ground truth annotations are frame derived and therefore not perfectly aligned with microsecond level event data, but this is not the primary issue. The more significant problem is that these annotations are generated by object detection algorithms that label all relevant objects in the scene, including those that are completely static. AMOS, on the other hand, is designed to segment only moving objects. As a result, the ground truth contains many bounding boxes that do not correspond to event generating motion, which means that standard evaluation metrics cannot be fully trusted in this context even though the metrics themselves are correct for datasets with accurate labels. Despite this limitation, consistent improvements across both datasets confirm that short-term depression offers an effective way to distinguish true motion from ego-motion and background noise.

The study also highlights several limitations and opportunities. DBSCAN performs reasonably well for grouping spatial clusters of events, but its sensitivity to parameters such as neighborhood radius and window duration limits full automation. Since the clustering step collapses temporal information into short windows, some fine grained motion cues are lost. A fully three dimensional approach that uses the x, y, and t coordinates of events may improve segmentation continuity. Similarly, the combination of polarity information could enhance selectivity for leading and trailing edges. Extending the spiking network with additional layers, or stacking depressing and facilitating connections, may allow multi scale representations of motion. Finally, the lack of event native annotations in current datasets limits evaluation quality and suggests a need for improved datasets.

In conclusion, AMOS demonstrates that short-term synaptic plasticity can serve as a biologically grounded and computationally efficient mechanism for motion segmentation in event-based vision. The depressing Tsodyks Markram synapse model, in particular, provides an adaptive filter that naturally suppresses sustained background activity while preserving meaningful transient motion signals. By operating entirely in an event driven manner, the framework avoids reconstruction, supervised training, and frame based processing. This work contributes an interpretable and biologically motivated approach to event-based perception and offers a foundation for future neuromorphic systems capable of real time, low power motion understanding in dynamic environments.

Acknowledgements

First and foremost, I would like to thank Stefano. He has been much more than an academic supervisor. He has been a mentor, a guide, and a genuine friend outside the university. I would not be here without his support, his patience, and his belief in me. My sincere thanks also go to Alessandro and Enrico, who were always there to answer my questions whenever I needed them.

I was surrounded by two Alessios throughout this thesis journey. I would like to thank Alessio Carpegna for introducing me to this fascinating topic and setting me on this path. I also want to thank Alessio Caviglia for being the most humble and insightful supervisor, for listening to all my boring and sometimes crazy ideas, and for challenging me with the right questions at the right moments.

I owe deep gratitude to my family: my father, my mother, and my sister. As always, they have been my biggest supporters, even though they have no idea what I am actually doing in this thesis. Their unconditional love has been my greatest strength.

They say that cities are shaped by the people we meet in them. I want to thank the people who made Torino one of the best chapters of my life. My heartfelt thanks go to Meric, my companion throughout this journey, with whom I faced every struggle and enjoyed every new adventure. To Selen and Oguzhan, the first and closest friends I made in this city. To Ozge, Ecem, and Koray, who turned every Friday night into a memory. To Federica, the perfect coffee partner during long afternoon study sessions. To Benedetto, the ideal cinema companion for films that nobody else wanted to watch. And to Goktug, my running buddy, with whom every run became a conversation about everything and nothing.

Life has taken us to different parts of the world, yet distance never changed anything. To Gulce, Utku, and Deniz, my dearest lifelong friends. Thanks to the time zones, at least one of you has always been there. We may be far from each other right now, but knowing you are only a call away makes the world feel much smaller and much kinder.

Finally, I want to thank Lab 6. You made me feel at home from the very first day. Thank you for being more than colleagues. Thank you for being true friends.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Biological Inspiration and Neuromorphic Motivation

Throughout the history of engineering, humans have repeatedly turned to nature for inspiration. Living organisms represent some of the most adaptive, resilient, and efficient systems known, capable of growing, healing, reproducing, and operating under extreme conditions. Early technological progress often began with simple observations of biological mechanisms. The attempts to build flying machines were initially inspired by the wings of birds, long before the principles of aerodynamics were formalized. Similar examples appear across many fields, from echolocation in bats influencing sonar systems to the visual system of insects inspiring agile robotics. In many cases, engineers first tried to imitate what they observed, and only later did scientific understanding reveal the underlying physical laws. This process of reverse engineering biological systems has shaped much of modern technological development.

In the age of machine learning and computational intelligence, the human brain represents one of the most complex and powerful biological systems to draw inspiration from. The brain combines processing, memory, and adaptation within the same substrate, operating at a level of efficiency and robustness far beyond conventional computing architectures. While experimental neuroscientists study the mechanisms of neural circuits through biological experimentation, engineers attempt to translate these principles into computational models and hardware that can solve real-world problems. Neuromorphic engineering lies at the intersection of these two domains. It aims to capture the essential principles of neural computation and implement them in artificial systems that benefit from the energy efficiency, adaptability, and event-driven nature observed in biological brains.

1.2 Event-Based Vision and Challenges

One domain where neuromorphic principles have gained significant traction is computer vision, particularly in dynamic environments. Conventional frame-based vision systems operate by capturing full images at a fixed rate, regardless of how much of the scene changes over time. This leads to redundant data, motion blur, and high computational cost, especially in fast or low-light scenarios. Event-based cameras, such as the Dynamic Vision Sensor (DVS) and Active Pixel Sensor (APS)-based sensors, provide an alternative approach. Instead of recording snapshots at fixed intervals, they detect and transmit only changes in pixel intensity. This produces an asynchronous stream of events with high temporal resolution and low redundancy. Event-based vision is therefore naturally aligned with Spiking Neural Networks (SNNs), which also operate on discrete events rather than continuous-valued signals.

Despite these advantages, event-based data presents its own challenges. Scenes captured from moving platforms such as cars generate large amounts of background activity due to ego-motion. This background activity mixes with events generated by independently moving objects such as vehicles, pedestrians, and cyclists. Filtering out ego-motion-induced events while preserving meaningful motion patterns remains a significant challenge, especially when object speeds, lighting conditions, and scene complexity vary over time. Existing neuromorphic and event-based approaches often make simplifying assumptions or rely on models that do not fully adapt to the structure of the input. Some methods treat temporal isolation alone as sufficient, while others require iterative optimization, frame reconstruction, or supervised training. As a result, they often struggle in realistic driving scenarios that involve continuous motion, overlapping objects, or non-uniform background activity.

1.3 Problem Statement

The central problem addressed in this thesis is how to isolate meaningful motion patterns from event-based camera data in a fully event-driven and biologically plausible manner. The objective is to suppress background activity generated by ego-motion while highlighting events that originate from independently moving objects. Achieving this requires a mechanism that adapts in real time to both the temporal structure and the local spatial patterns of the event stream. Such a mechanism must be robust to variations in motion speed, responsive to transient features, and computationally efficient enough to operate on large-scale datasets.

1.4 Proposed Approach

To address this problem, this thesis proposes a neuromorphic motion segmentation framework based on SNNs with STP. Inspired by synaptic dynamics in biological neural circuits, the approach uses Tsodyks Markram (TM) synapses to modulate the efficacy of synaptic transmission according to recent presynaptic activity. Depressing synapses reduce their influence during sustained activation, which naturally suppresses repetitive background events caused by ego-motion. Facilitating synapses enhance their influence during repeated activation, which emphasizes consistent motion patterns associated with independently moving objects. Combined with spatially local convolution-like receptive fields, this synaptic adaptation enables the network to perform adaptive filtering directly on the event stream without training or reconstruction.

After filtering, the remaining events are grouped into spatially coherent structures using clustering techniques. Because the number and size of objects vary over time, and because event-based data naturally form irregular spatial patterns, density-based clustering, such as Density Based Spatial Clustering of Applications with Noise (DBSCAN), is well suited to this task. The clustering stage converts spike activity into bounding box predictions that can be compared with ground truth annotations.

1.5 Contributions

The main contributions of this thesis are as follows.

- A biologically grounded framework for event-driven motion segmentation based on the combination of local spatial filtering and STP.
- A systematic analysis of how different neuron and synapse configurations influence motion filtering, including static, depressing, and hybrid synaptic dynamics.
- A demonstration that depressing TM synapses provide a robust and adaptive mechanism for suppressing ego-motion-induced background activity in realistic driving environments.
- An integrated pipeline that connects NEural Simulation Tool (NEST)-based spiking simulation with post processing and clustering to produce full bounding box predictions on large automotive datasets.
- A comprehensive evaluation on both the MVSEC and Prophesee 1 Megapixel Automotive Datasets, showing that biologically inspired synaptic adaptation can improve event-driven motion segmentation without supervised learning or heavy computational models.

1.6 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 provides the necessary background on neuromorphic computing, SNNs, event-based vision, and object detection in dynamic environments. Chapter 3 describes the methodology, including the NEST simulation environment, the proposed filtering models, and the clustering process. Chapter 4 presents the experimental setup, results, and discussion. The thesis concludes with a summary of findings and potential directions for future work.

Chapter 2

Background

2.1 Neuromorphic Computing: From Biology to Engineering

Modern computation is fundamentally built upon the von Neumann architecture, proposed in 1945, which physically separates processing and memory units. In this structure, data must shuttle continuously between a central processor and an external memory via a limited communication bus. Although this design has underpinned decades of digital progress, it has also introduced an inherent inefficiency known as the von Neumann bottleneck, where data transfer becomes the main source of latency and energy consumption. As applications grow increasingly data-intensive, from large-scale machine learning to pervasive sensor networks, these constraints become critical. It is estimated that 5–15 % of the world's total energy is already consumed by data movement and computation [1]. Future exascale supercomputers, if still based on this paradigm, are projected to draw tens of megawatts of power, far exceeding sustainable limits. Moreover, conventional processors lack the adaptive, self-organizing properties characteristic of biological intelligence, relying instead on explicit programming and sequential instruction flow.

In contrast, the human brain exemplifies an entirely different computational principle. Biological neurons integrate information, store state, and perform computation within the same local structure, effectively merging memory and processing. These neurons communicate using spikes—discrete, time-based electrical pulses that encode information in both timing and rate. Through synaptic plasticity, the strength of connections between neurons evolves as a function of activity, allowing the system to learn and adapt continuously. Despite operating with roughly 10^{11} neurons and 10^{15} synapses, the brain consumes only about a few watts of power, several orders of magnitude more efficient than even the most advanced supercomputers. Its architecture is inherently event-driven, massively parallel, and fault-tolerant, where computation occurs only when needed. These properties form the inspiration for neuromorphic computing, a discipline that aims to replicate the efficiency, adaptability, and resilience of biological systems using electronic devices [1, 2].

The concept of neuromorphic engineering originated with Carver Mead in the late 1980s, who pioneered the use of analog VLSI circuits to emulate the physical behavior of

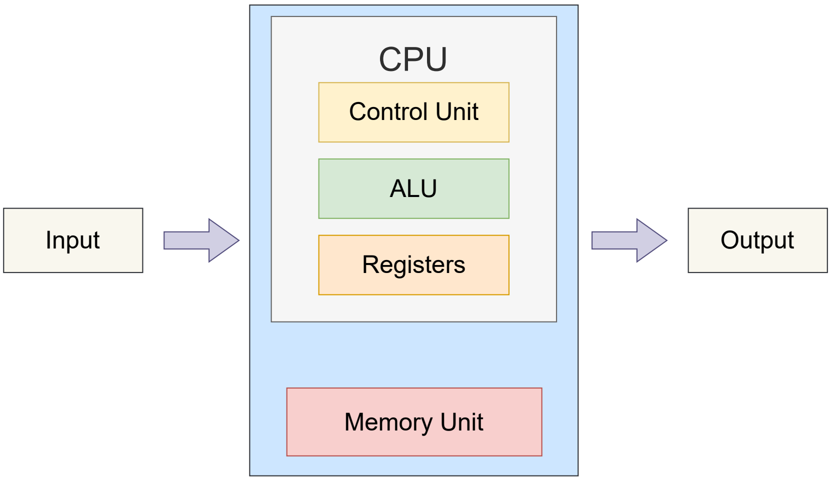


Figure 2.1: Von Neumann architecture

neurons and synapses. Mead's seminal work [3] inspired early prototypes such as the silicon retina and silicon cochlea, which mimicked biological sensory processing through analog circuits operating in the subthreshold regime [4]. These early systems demonstrated that biological principles such as local processing and temporal encoding could be realized in silicon, introducing a new paradigm distinct from both traditional digital computing and software-based neural networks.

Over the years, the definition of neuromorphic computing has expanded beyond purely analog systems. Today, it encompasses digital, mixed-signal, and hybrid architectures that implement brain-inspired computation in hardware. Modern neuromorphic platforms retain key biological principles:

- Event-driven operation, where computation occurs only in response to input spikes, minimizing idle energy.

- Co-location of memory and computation, reducing data movement by storing synaptic weights locally.

- Massive parallelism, achieved through thousands of concurrently active neurons and synapses.

- Asynchronous communication, which eliminates global synchronization and enhances scalability.

These properties enable neuromorphic systems to process temporal and sensory information with remarkable energy efficiency. As Christensen [1] note, such systems are particularly promising for edge computing, where low-power, on-device intelligence is essential for autonomous perception and control.

The implementation of neuromorphic principles has evolved through a wide range of hardware and simulation platforms, each emphasizing different trade-offs between biological realism, computational scalability, and power efficiency.

IBM TrueNorth (2014) demonstrated large-scale digital neuromorphic integration with 1 million neurons and 256 million synapses, achieving energy efficiencies around 70 mW per million neurons. Its architecture emphasizes deterministic spike routing and high-throughput parallel inference [5].

Intel Loihi introduced in 2018, extends this paradigm by enabling on-chip learning. Each Loihi core supports programmable plasticity mechanisms such as Spike-Timing-Dependent Plasticity (STDP) and reinforcement learning, facilitating online adaptation [6]. Loihi's asynchronous, mesh-based interconnect allows distributed real-time processing without centralized control.

SpiNNaker developed at the University of Manchester, implements a fully digital, packet-based communication network interconnecting thousands of ARM cores. It can simulate up to a billion neurons in real time, providing a testbed for large-scale cortical models [1].

BrainScaleS from Heidelberg University, adopts a hybrid analog-digital approach that emulates membrane potential dynamics directly in hardware. By operating 10^4 times faster than biological time, it enables accelerated experimentation on neural dynamics and learning mechanisms [1].

NEST as complementing these physical systems, serves as a cornerstone of software-based neuromorphic research. Designed for high-performance computing environments, NEST provides a flexible framework for simulating large-scale spiking neural networks with configurable neuron, synapse, and plasticity models. It supports both biologically detailed simulations and large-scale computational studies across distributed nodes. NEST's open architecture has made it a reference tool for neuroscience and neuromorphic engineering alike, often used to prototype network dynamics or test new synaptic models before deployment on physical neuromorphic hardware [7].

Together, these systems represent a continuum from biology to technology. TrueNorth and Loihi emphasize low-power embedded intelligence; SpiNNaker and BrainScaleS focus on brain-scale modeling; and NEST provides the simulation backbone that bridges theoretical neuroscience with hardware implementation. This multi-level ecosystem reflects the broader trend in neuromorphic research: integrating device-level innovation, circuit design, and computational modeling into a unified framework for understanding and replicating neural computation.

As highlighted in recent reviews, the field has matured beyond prototype systems developed in academic laboratories into production-level infrastructures with event-driven processing, learning models, and community-driven software ecosystems [8]. These advances collectively indicate that neuromorphic computing is entering a critical phase, focused on scaling principles, interoperability, and standardized toolchains to ensure broad adoption. At its core, however, the success of neuromorphic computing depends on the neural model that governs how spikes are generated, transmitted, and integrated. Whether implemented in hardware or simulated in software, these systems rely on abstractions of biological neurons that capture the dynamics of spiking, membrane potential evolution, and synaptic transmission. The next section explores this computational foundation in depth through the lens of SNNs the algorithmic framework that lies at the heart of all neuromorphic architectures. SNNs provide the mathematical and functional description of how information is encoded in time, how synaptic plasticity shapes learning, and how event-driven processing leads to energy-efficient intelligence.

2.2 Spiking Neural Networks

At the algorithmic core of neuromorphic computing lie SNNs, which extend the concept of conventional artificial neurons by incorporating time as a fundamental variable of computation. Unlike Artificial Neural Networks (ANNs), which exchange continuous activation values at fixed time steps, SNNs transmit information through discrete events known as spikes. These spikes, represented as binary signals, encode information not only in their occurrence but also in their precise timing. This temporal dimension allows SNNs to process dynamic, asynchronous data streams and to perform computation only when necessary, offering a pathway toward highly energy-efficient and event-driven intelligence [9, 10, 11]. Beyond their engineering advantages, SNNs also serve as computational abstractions of biological neural systems, linking neuroscience and machine learning under a unified mathematical framework.

2.2.1 Spiking Neuron Models

Biological neurons act as the basic computational units of the brain. They receive inputs through dendrites, integrate these signals in the soma, and emit an electrical pulse, or spike, when the membrane potential surpasses a threshold. This process can be described as a continuous evolution of the membrane potential, shaped by both external currents and intrinsic electrical properties of the cell. Spiking neuron models abstract this biophysical process into computational form, enabling simulation and mathematical analysis while retaining the essential mechanism of integration and firing.

The simplest and most fundamental abstraction is the Integrate-and-Fire (IF) model. In this formulation, the neuron integrates incoming weighted inputs over time, and when the accumulated potential crosses a threshold V_{th}, a spike is generated. Immediately after firing, the potential is reset to a resting value E_L. The temporal evolution of the membrane potential $V_m(t)$ can be expressed as:

$$\frac{dV_m(t)}{dt} = \frac{I(t)}{C_m}, \tag{2.1}$$

where $I(t)$ is the total synaptic input current and C_m is the membrane capacitance. When $V_m(t) \geq V_{th}$, the neuron emits a spike and resets V_m to E_L. Despite its simplicity, the IF model captures the essential threshold-based firing mechanism and is computationally efficient, making it suitable for large-scale simulations where millions of neurons are modeled. However, this abstraction neglects the passive decay of membrane voltage that naturally occurs in biological neurons, meaning that the potential would increase indefinitely in the presence of sustained input. As a result, the IF model fails to capture the temporal filtering properties observed in real neurons, where past inputs gradually lose influence if not reinforced by new spikes [10, 11].

To address this limitation, the Leaky Integrate-and-Fire (LIF) model introduces a passive decay term that continuously drives the membrane potential back toward its resting state. This leak mimics the diffusion of ions through the membrane in biological cells and ensures that the neuron responds primarily to recent inputs rather than accumulating all past activity. The dynamics of the LIF neurons are described by the following differential equation:

$$\tau_m \frac{dV_m(t)}{dt} = -(V_m(t) - E_L) + RI(t), \tag{2.2}$$

where $\tau_m = RC_m$ is the membrane time constant, E_L is the resting potential, and R is the membrane resistance. When the potential reaches the threshold V_{th}, the neuron emits a spike and resets to E_L. The inclusion of the leak term adds only minimal computational overhead compared to the pure IF model, yet it dramatically improves biological plausibility by producing temporally stable and decaying membrane dynamics. For this reason, the LIF model has become the standard choice in most neuromorphic simulators, such as NEST and Brian, as well as in hardware implementations like Intel's Loihi chip. A comparative figure illustrating the membrane potential trajectories of IF and LIF neurons would show that, whereas the IF neuron's potential increases linearly until spiking, the LIF neuron exhibits exponential decay between synaptic inputs, resulting in more biologically realistic firing behavior.

Beyond the LIF model, several extensions further increase biological fidelity at the cost of computational complexity. The *Adaptive Exponential Integrate-and-Fire (AdEx)* model introduces an adaptation current that dynamically modulates the firing threshold, reproducing bursting, spike-frequency adaptation, and other temporal firing patterns seen in cortical neurons. Similarly, the *Izhikevich model* offers a compact yet flexible formulation capable of capturing a wide spectrum of neuronal behaviors using a small set of parameters. These enhanced models provide valuable realism when studying specific neural phenomena but require solving additional differential equations and maintaining more internal state variables. Consequently, while IF and LIF neurons are preferred for efficiency in large-scale or hardware-oriented simulations, models like AdEx and Izhikevich are favored when biological interpretability and detailed spiking dynamics are the priority [10, 11].

In summary, the progression from IF to LIF and then to more complex adaptive models illustrates a fundamental trade-off in neuromorphic modeling: increasing biological realism typically comes at the expense of computational simplicity. Depending on the purpose large-scale simulation, real-time hardware implementation, or detailed biological study, the choice of neuron model must balance accuracy, efficiency, and interpretability.

The behavior of individual spiking neurons forms the foundation for how information

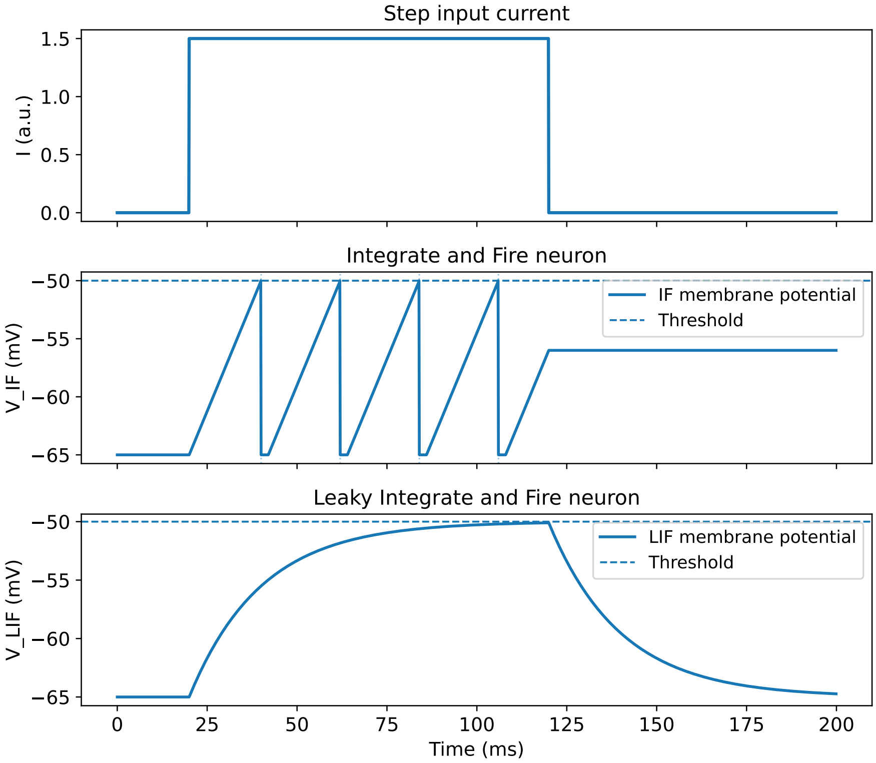


Figure 2.2: Comparison of Integrate and Fire and Leaky Integrate and Fire neurons under the same step current input

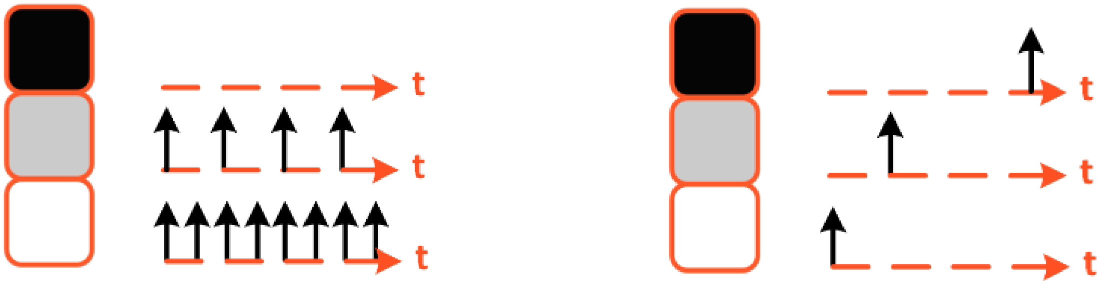
is represented and processed in spiking neural networks. Once spikes are generated, their timing, frequency, and correlation across neurons become the medium through which signals are transmitted and encoded. Understanding these temporal patterns is essential to explaining how SNNs store and transform information, leading naturally to the discussion of temporal coding and information representation in the next section.

2.2.2 Temporal Information Processing and Coding Schemes

Information representation in SNNs is inherently temporal. Unlike conventional neural networks, where neurons transmit static activation values, spiking neurons communicate using discrete events that occur in time. Consequently, the meaning of a neural signal is not embedded in its amplitude, but rather in the timing, frequency, and correlation of spikes. The way these spikes encode information, referred to as the *neural coding scheme*, is fundamental to how an SNN learns and processes data. Choosing a coding scheme is therefore not a mere implementation detail; it directly influences the learning algorithm, the network's energy efficiency, and its ability to represent dynamic sensory inputs. Figure 2.3 comparing different spike-based coding schemes can visually highlight these distinctions and their implications for computation.

The most widely used approach is *rate coding*. In this scheme, information is encoded in the average firing rate of a neuron within a specific time window. The higher the frequency of spikes, the stronger the represented signal. Rate coding is conceptually simple and biologically supported by early studies of sensory neurons, where firing rates correlate with stimulus intensity. From a computational perspective, rate coding allows the use of statistical averaging, making it robust to individual spike noise and compatible with many conversion-based training methods that map continuous activations to spike rates. However, this robustness comes at a cost. To estimate a meaningful rate, neurons must emit a sufficient number of spikes over time, which requires long simulation windows and results in higher energy consumption. This can be illustrated by showing how a rate-coded neuron accumulates multiple spikes to approximate a static signal level, sacrificing temporal precision for stability.

In contrast, *temporal coding* encodes information in the exact timing or relative latency of spikes. Here, a single spike may be sufficient to convey a meaningful message if its timing is precise. For instance, in a latency-coded scheme, the time delay between stimulus onset and the first emitted spike reflects the stimulus intensity—stronger inputs produce earlier spikes. Temporal coding aligns more closely with biological observations in auditory and visual cortices, where neurons synchronize precisely to external rhythms or motion cues. Computationally, this enables faster and more energy-efficient representations, since fewer spikes are needed to transmit information. Yet, this precision makes temporal coding more sensitive to noise and hardware jitter, posing challenges for stability and learning. A suitable figure could show how neurons with temporal coding respond almost instantaneously to stimuli, achieving high responsiveness with sparse spiking activity.



(a) Rate coding: An input pixel of greater intensity corresponds to a higher firing rate

(b) Temporal coding: An input pixel of greater intensity corresponds to an earlier spike time.

Figure 2.3: Most common coding schemes [12]

While rate and temporal coding represent two extremes as averaged versus time-precise signaling, hybrid schemes combine their advantages. *Phase coding*, for example, uses spike timing relative to an oscillatory phase, and *population coding* distributes information across ensembles of neurons to enhance robustness and redundancy. These strategies balance biological plausibility and computational reliability, often appearing in networks designed for event-based vision or sensory fusion. Importantly, as noted by Yi et al. [10], neural coding strategies enable both the brain and artificial spiking systems to represent spatiotemporal patterns compactly and adaptively. The chosen coding scheme determines how information flows through layers, how learning rules operate, and how effectively the network can exploit temporal structure. Therefore, the decision between rate-based and time-based representations is not just architectural, it can fundamentally reshape the algorithmic behavior and performance of the entire spiking system.

2.2.3 Training Approaches for Spiking Neural Networks

Training SNNs remains one of the main challenges in neuromorphic computing because the process of generating spikes is inherently non-differentiable. Unlike conventional ANNs, where activations are continuous and gradients can be computed directly, the binary and discontinuous nature of spikes prevents the straightforward application of standard backpropagation algorithms. Over the years, several training strategies have been developed to address this issue, each reflecting a different trade-off between biological realism, computational cost, and learning efficiency. These strategies can be grouped into four main categories: conversion-based training, direct gradient-based optimization, biologically inspired local learning rules, and evolutionary or reinforcement-based optimization [10, 11, 13, 9].

Conversion-based Training

Conversion-based training was one of the earliest approaches proposed for SNNs. Rather than training a spiking model directly, a conventional ANN is first optimized using standard gradient-descent techniques, after which its learned parameters are transferred to an equivalent spiking architecture by mapping continuous activations to firing rates. In this way, the spiking network can reproduce the behavior of the original ANN while exploiting the sparse and event-driven computation that characterizes SNNs.

This method is conceptually simple and compatible with existing deep-learning models, allowing researchers to reuse architectures such as VGG or ResNet. It is particularly advantageous for inference-oriented applications, where energy efficiency and latency are prioritized over online adaptation. However, since the temporal dynamics of spiking neurons are not explicitly modeled during training, converted networks typically rely on rate coding, which requires longer simulation windows to estimate firing statistics accurately. This limitation can increase inference latency and energy consumption. Additionally, the transformation from continuous activations to spike rates introduces information loss and limits the precision of temporal processing [13].

Recent methods such as knowledge-distillation-based conversion mitigate these drawbacks by transferring not only the final outputs but also intermediate feature representations from a trained ANN (the teacher) to a spiking network (the student). This process enhances both the representational richness and robustness of the converted network, enabling SNNs to achieve competitive accuracy on large-scale visual tasks while maintaining low power consumption [13, 9].

Direct Gradient-based Training

Direct gradient-based training methods were developed to overcome the limitations of indirect conversion and to optimize spiking networks directly within the temporal domain. The central idea is to apply Backpropagation Through Time (BPTT), a technique commonly used for recurrent neural networks, to capture the evolution of membrane potentials and spike activity over time. During training, the network is unfolded across discrete time steps, and gradients are propagated backward through each neuron's membrane potential updates to minimize a task-specific loss function [9, 11].

The primary challenge in this approach arises from the spike generation mechanism itself, which is defined by a discontinuous threshold function. Its derivative is zero almost everywhere and undefined at the firing point, making it impossible to compute exact gradients. To address this, *surrogate gradient* methods replace the non-differentiable spike function with a smooth approximation during the backward pass. Common surrogate functions include sigmoid, fast-sigmoid, exponential, or piecewise-linear shapes that approximate the sharp threshold transition while maintaining differentiability [9].

This approximation provides a non-zero slope around the threshold, allowing learning signals to propagate backward through the spiking network. During the forward pass, the neuron still emits binary spikes, preserving the discrete dynamics of biological neurons, while during the backward pass, the smooth surrogate function enables efficient gradient computation. The surrogate gradient approach has become the foundation of modern SNN optimization and is implemented in frameworks such as SLAYER, snnTorch, and Norse.

Variants like the e-prop algorithm further improve scalability by estimating local gradients without storing the entire network history, significantly reducing memory usage and computational cost. Although direct gradient-based training methods achieve state-of-the-art performance on event-based benchmarks and enable fully supervised learning, they remain computationally intensive and less biologically plausible than local learning rules [9, 10]. Nevertheless, they provide an effective bridge between traditional deep-learning techniques and neuromorphic computation, demonstrating that spiking networks can be trained using modern optimization principles while preserving temporal processing capabilities.

Biologically Inspired Local Learning Rules

While gradient-based learning allows SNNs to achieve high performance on benchmark tasks, biological neural systems rely on local adaptation mechanisms that depend only on spike interactions. Instead of propagating global error signals, neurons modify their synaptic connections based on locally available information such as pre- and post-synaptic firing activity. In spiking neural networks, these mechanisms are modeled through *synaptic plasticity rules* that govern how the synaptic weight w evolves over time as a function of spike timing, firing rate, and synaptic state variables [10, 11, 14, 15].

The foundational concept underlying these learning mechanisms is the *Hebbian principle*, often summarized as "neurons that fire together wire together." It states that when a presynaptic neuron consistently contributes to the activation of a postsynaptic neuron, their synaptic connection strengthens. A simple mathematical expression of this principle can be written as:

$$\Delta w = \eta \, x_{\text{pre}} \, x_{\text{post}}, \tag{2.3}$$

where η is the learning rate, and x_{pre} and x_{post} denote the activity of the pre- and post-synaptic neurons, respectively. This local correlation rule forms the basis for both short-term and long-term plasticity.

Short-Term Plasticity (STP): *Short-Term Plasticity (STP)* refers to transient, reversible changes in synaptic efficacy that occur on the timescale of milliseconds to seconds. Unlike long-term mechanisms that permanently alter the synaptic weight, STP modulates the *effective strength* of a synapse dynamically according to recent presynaptic activity. STP manifests as two complementary processes: *facilitation*, which temporarily increases neurotransmitter release probability after repeated firing, and *depression*, which decreases it due to vesicle depletion.

A general mathematical description of STP can be formulated as:

$$\frac{du}{dt} = -\frac{u - U}{\tau_f} + U(1 - u) \sum_k \delta(t - t_k^{\text{pre}}), \tag{2.4}$$

$$\frac{dx}{dt} = \frac{1 - x}{\tau_d} - ux \sum_k \delta(t - t_k^{\text{pre}}), \tag{2.5}$$

where $u(t)$ represents the utilization of synaptic resources (facilitation), $x(t)$ the fraction of available neurotransmitters (depression), U the baseline utilization, τ_f and τ_d the facilitation and depression time constants, and t_k^{pre} the timing of presynaptic spikes. The effective postsynaptic response is then defined as:

$$A(t) = w \, u(t) \, x(t). \tag{2.6}$$

Through this mechanism, STP acts as a *temporal filter*: it emphasizes novel or rapidly changing inputs while suppressing repetitive patterns. In neuromorphic systems, STP provides a biologically inspired means of implementing adaptive signal processing without modifying long-term weights, which makes it particularly useful for event-based and streaming sensory data.

Long-Term Plasticity (LTP and LTD): *Long-Term Plasticity (LTP/LTD)* describes persistent modifications of synaptic strength that underlie learning and memory in the brain. When presynaptic and postsynaptic neurons repeatedly exhibit correlated activity, their synaptic connection strengthens (LTP), whereas uncorrelated or anti-correlated activity weakens it (LTD). A general form of this process can be represented as:

$$\frac{dw}{dt} = \eta \, F(\text{pre}, \text{post}), \tag{2.7}$$

where $F(\text{pre}, \text{post})$ denotes a nonlinear correlation function describing how pre- and post-synaptic firing patterns influence synaptic modification. Positive correlation ($F > 0$) leads to potentiation, and negative correlation ($F < 0$) results in depression.

STDP: A precise and widely studied formulation of long-term plasticity is STDP, which explicitly links the timing between pre- and post-synaptic spikes to the direction and magnitude of synaptic modification. If the presynaptic neuron fires shortly before the postsynaptic neuron, the connection is strengthened (LTP). Conversely, if the postsynaptic spike precedes the presynaptic one, the connection is weakened (LTD). This relationship can be modeled mathematically as:

$$\Delta w = \begin{cases} A_+ e^{-\Delta t/\tau_+}, & \text{if } \Delta t > 0, \\ -A_- e^{\Delta t/\tau_-}, & \text{if } \Delta t < 0, \end{cases} \tag{2.8}$$

where $\Delta t = t_{\text{post}} - t_{\text{pre}}$ is the temporal difference between post and presynaptic spikes, A_+ and A_- are the maximum potentiation and depression amplitudes, and τ_+ and τ_- represent their respective decay constants. This formulation unifies LTP and LTD as two complementary outcomes of a single timing-dependent learning rule [10, 11, 14].

More advanced extensions of STDP introduce additional modulatory factors that influence weight updates, leading to *three-factor learning rules*. These can be expressed as:

$$\Delta w = \eta\, M(t)\, G(\text{pre}, \text{post}), \tag{2.9}$$

where $M(t)$ is a global modulatory signal, such as a reward or dopamine concentration, and $G(\text{pre}, \text{post})$ describes local spike-based interactions. These mechanisms provide a bridge between unsupervised and reinforcement learning by allowing synaptic changes to be shaped by both local correlations and global feedback signals.

Together, short- and long-term plasticity mechanisms form a hierarchical and biologically plausible framework for local learning in SNNs. STP enables rapid adaptation to changing stimuli by modulating synaptic efficacy on short timescales, while STDP implements lasting memory traces through spike-timing-dependent weight changes. Their integration allows spiking neural networks to learn continuously and autonomously, adapting to dynamic environments without relying on global error propagation.

Evolutionary and Reinforcement-based Optimization

A complementary class of training methods for SNNs treats learning as a global optimization process rather than relying on gradient propagation. These approaches explore the large and non-differentiable parameter space of spiking networks, including synaptic weights, firing thresholds, delays, and even connectivity patterns. Since they do not depend on differentiable operations, they are particularly suitable for neuromorphic systems where discrete spikes, hardware constraints, or non-continuous dynamics make backpropagation impractical [9, 10, 11].

Evolutionary algorithms such as genetic algorithms, differential evolution, and particle swarm optimization employ population-based search strategies to iteratively improve candidate solutions according to a fitness function. Each generation refines the population through selection, mutation, and recombination, enabling the discovery of optimal or near-optimal configurations. These algorithms can optimize both network topology and synaptic parameters, supporting multi-objective goals such as accuracy, sparsity, and energy efficiency. Their main advantage lies in their ability to explore complex, non-convex optimization landscapes and adapt model structure autonomously. However, evolutionary methods are computationally demanding because they require evaluating many candidate networks over multiple generations. Hybrid strategies that combine evolutionary search for network structure with gradient-based fine-tuning have been proposed to mitigate this limitation [16].

Reinforcement-based optimization draws inspiration from reward-driven learning observed in biological systems, where synaptic changes are guided by global feedback signals rather than explicit error gradients. A widely used formulation is the *reward-modulated spike-timing-dependent plasticity (R-STDP)* rule, which extends STDP by introducing a scalar reward term that scales weight updates according to task performance:

$$\Delta w = \eta \, R(t) \, f(\Delta t), \tag{2.10}$$

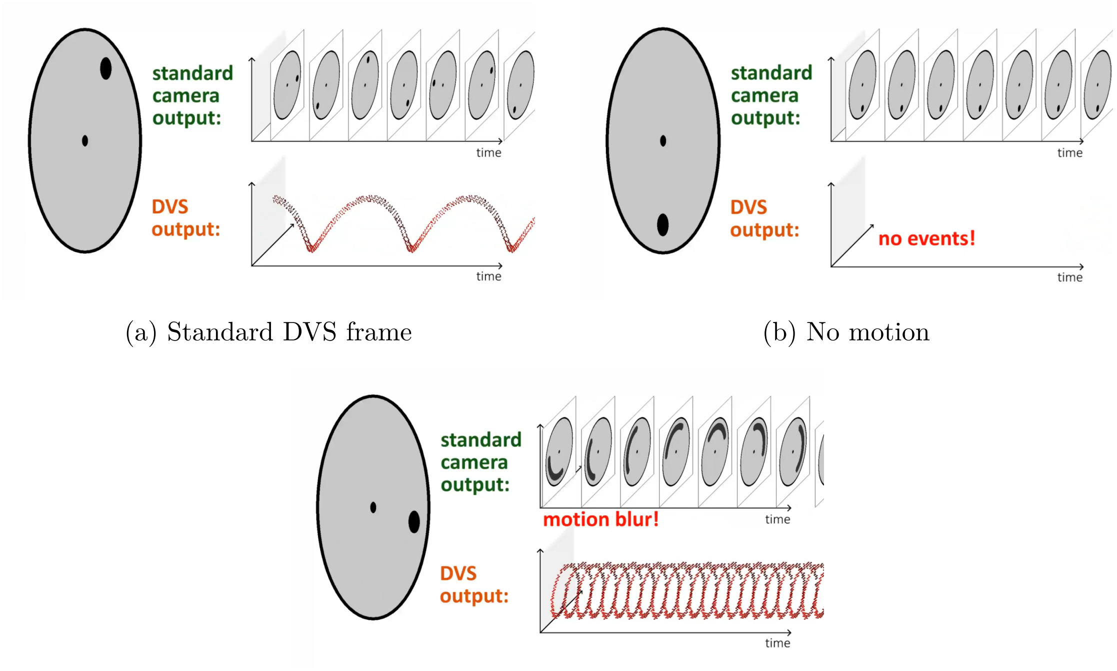
where $R(t)$ denotes the reward signal and $f(\Delta t)$ represents the STDP learning kernel as a function of spike timing. This mechanism allows networks to associate spiking activity with favorable outcomes, enabling adaptive behavior in control or decision-making tasks. Although biologically plausible and compatible with neuromorphic hardware, reinforcement-based methods generally converge slowly and scale poorly for large or deep architectures [10, 14].

In summary, evolutionary and reinforcement-based optimization represent biologically inspired, gradient-free alternatives to supervised learning. They are robust to discontinuities, require no backpropagated gradients, and are well suited for on-chip or online learning scenarios. However, their computational expense and limited scalability remain active challenges. Recent research increasingly explores hybrid frameworks that integrate these global optimization strategies with local plasticity or gradient-based learning to balance adaptability, efficiency, and biological realism.

2.3 Event-Based Vision

Conventional frame-based cameras capture visual information at fixed time intervals, typically 30 or 60 frames per second. Each frame represents a complete snapshot of the scene, regardless of whether changes occur in every pixel. This approach, while effective for many applications, introduces limitations in dynamic or high-speed environments. The discrete sampling process leads to temporal aliasing and motion blur when objects move faster than the frame rate, and the redundant capture of static regions results in massive data overhead. Additionally, because every frame requires global exposure and readout, traditional cameras exhibit limited dynamic range and high latency, especially under challenging lighting or motion conditions [17, 18].

Event-based cameras, also known as neuromorphic or bio-inspired vision sensors, operate under a fundamentally different principle. Instead of recording full images at fixed



(a) Standard DVS frame

(b) No motion

(c) Blurred scene

Figure 2.4: Behaviors of standard camera output and event-based camera output in different scenarios [19].

intervals, they detect changes in brightness asynchronously and independently at each pixel. Whenever the logarithmic intensity at a pixel changes by more than a predefined contrast threshold, the pixel generates an event, encoding the occurrence, location, time, and polarity of that change. This mechanism, first implemented in the Dynamic Vision Sensor (DVS) and later extended in devices such as DAVIS and Prophesee, draws direct inspiration from biological retinas, which signal only local luminance changes rather than absolute intensity values [20, 18].

The benefits of this sensing paradigm are significant. Event cameras achieve microsecond-level temporal resolution and sub-millisecond latency because each pixel operates independently and continuously. They capture motion without blur, maintain wide dynamic ranges exceeding 120 dB, and drastically reduce redundant data by transmitting only meaningful brightness variations. These properties lead to lower bandwidth and power consumption, making event-based sensors ideal for embedded and edge computing. Such characteristics are particularly advantageous in robotics, autonomous vehicles, and surveillance systems, where real-time and energy-efficient processing is critical [17].

Event-based vision therefore provides a sensory front-end naturally aligned with the temporal and sparse information processing capabilities of spiking neural networks. Both systems share asynchronous, event-driven computation and temporal precision, forming a biologically consistent integration between sensing and processing.

2.3.1 Asynchronous Sensing Principles

At the core of event-based sensing lies its asynchronous, data-driven operation. Each pixel measures the logarithmic light intensity $L(x, y, t) = \log I(x, y, t)$, where I represents the input brightness. An event is triggered when the change in intensity exceeds a threshold C:

$$\Delta L(x, y, t_k) = L(x, y, t_k) - L(x, y, t_{k-1}) \geq pC, \tag{2.11}$$

where $p \in \{+1, -1\}$ denotes the polarity, indicating whether brightness increased (ON event) or decreased (OFF event). Each event can be represented as a tuple:

$$e_k = (x_k, y_k, t_k, p_k), \tag{2.12}$$

defining the spatial coordinates (x_k, y_k), timestamp t_k, and polarity p_k.

This event stream encodes dynamic information directly rather than static intensity values. Since each pixel operates independently, the number of generated events naturally adapts to scene dynamics: moving objects produce dense streams of events, while static regions remain inactive. From a signal-processing perspective, event streams approximate the temporal derivative of image intensity, emphasizing changes and eliminating redundancy.

The asynchronous nature of this operation removes the need for a global exposure time, allowing microsecond temporal precision and continuous motion capture. This principle of "only sensing change" reduces latency, increases temporal fidelity, and minimizes energy consumption.

Event-based sensing closely mirrors the biological retina, where photoreceptors and ganglion cells respond to changes in luminance rather than constant illumination. Similarly, event cameras generate spike-like outputs that resemble neural activity, making their data format inherently compatible with the temporal coding and sparsity of spiking neural networks [20, 18].

2.3.2 Event Cameras and Data Characteristics

Several commercial and research-grade event cameras are now widely used, including the Dynamic Vision Sensor (DVS), the Dynamic and Active Vision Sensor (DAVIS), and Prophesee's GEN series and 1-Megapixel sensors. The DAVIS family integrates both asynchronous DVS and conventional Active Pixel Sensor (APS) circuits, allowing the simultaneous capture of event and frame data streams. The Prophesee 1-Megapixel sensor and DAVIS346 models are among the most commonly adopted, offering microsecond-level temporal precision and high dynamic ranges beyond 120 dB [20].

Event-based datasets have expanded substantially, supporting a range of research domains. Examples include *MVSEC*, which provides multimodal data (stereo, IMU, LiDAR) for 3D perception; *DDD17* and *DSEC*, which target driving scenarios; *GEN1* and the *1-Megapixel Automotive Dataset* for object detection; and gesture recognition datasets such as *DVS Gesture* and *EHWGesture*, the latter combining RGB, depth, and event modalities for multimodal gesture understanding [21]. These datasets collectively cover applications in autonomous driving, robotics, and human–computer interaction, providing valuable benchmarks for both neuromorphic and conventional vision models.

Event data consist of discrete events $\{e_k = (x_k, y_k, t_k, p_k)\}$ collected over time. For processing, events can be grouped into time intervals to form event frames, voxel grids, or time surfaces, enabling the reuse of conventional convolutional pipelines. However, such accumulation partially sacrifices the asynchronous nature of the data. Alternatively, spiking neural networks and neuromorphic processors can operate directly on raw event streams, exploiting their temporal precision and sparsity for real-time and energy-efficient computation [17, 18].

Event-based data are characterized by high temporal density in dynamic regions, sparse spatial activity in static areas, and explicit polarity encoding. These attributes make event cameras ideal for applications that require low latency, high dynamic range, and motion awareness. By coupling event-based sensing with neuromorphic computation, complete perception pipelines can be built that operate efficiently, adaptively, and in real time, emulating the sensing–processing synergy observed in biological vision systems.

2.4 Object Detection in Dynamic Environments

Object detection is a fundamental task in computer vision, aiming to identify and localize objects within a scene by predicting their class labels and bounding boxes. Traditional frame-based approaches such as R-CNN, SSD, and YOLO have achieved remarkable success in static imaging domains, leveraging deep convolutional networks to perform large-scale detection across diverse categories. These architectures, operating on fixed-rate video or image sequences, have enabled breakthroughs in applications ranging from surveillance to autonomous driving. However, their reliance on frame-based acquisition and dense pixel processing introduces inefficiencies when applied to dynamic or resource-constrained environments. Each frame contains significant redundancy, as most pixels remain unchanged between consecutive captures. Moreover, motion blur, limited dynamic range, and high computational load hinder their performance under fast motion or rapidly changing illumination [22, 23].

Neuromorphic vision offers a promising alternative to overcome these limitations. Event-based sensors and spiking neural networks operate asynchronously, capturing only meaningful changes in the visual field. Instead of processing entire images at a fixed rate, they encode sparse, time-resolved events triggered by local brightness variations. This paradigm enables continuous perception with minimal latency and power consumption, as computation is driven solely by informative input rather than periodic sampling. In contrast to frame-based models, which often expend resources processing static background information, neuromorphic systems react instantaneously to motion, offering a scalable and energy-efficient solution for real-time perception in dynamic environments. Such efficiency makes them particularly well suited for mobile and embedded systems, where computational and energy budgets are limited [20, 18, 17].

2.4.1 Conventional Vision vs. Neuromorphic Vision

The contrast between conventional and neuromorphic vision highlights the trade-offs between temporal density, redundancy, and responsiveness. Frame-based systems acquire complete images at discrete intervals, operating under global synchronization. This design ensures compatibility with standard deep-learning pipelines but inherently introduces latency proportional to the frame rate. Additionally, each frame is processed in its entirety, regardless of scene dynamics, leading to significant data redundancy and unnecessary computation. For instance, in a surveillance camera observing a mostly static scene, identical background pixels are repeatedly analyzed, consuming bandwidth and energy without contributing new information [22, 23].

Neuromorphic vision, in contrast, replaces this periodic sampling with continuous, event-driven sensing. Each pixel operates independently, generating events only when the local logarithmic intensity changes beyond a defined threshold. This asynchronous mechanism ensures that processing resources are devoted exclusively to regions undergoing change. The resulting output is temporally precise, sparse, and free of redundant information, enabling microsecond response times. Beyond latency reduction, event-based systems exhibit resilience to motion blur and perform reliably under extreme lighting conditions where conventional sensors would saturate. These attributes make neuromorphic cameras advantageous for real-world applications requiring both speed and adaptability, including robotics, UAV navigation, and autonomous driving [18, 24].

While deep-learning methods in frame-based vision continue to evolve with architectures such as vision transformers and multi-scale detection heads, they remain constrained by the frame-based paradigm. Neuromorphic systems, by contrast, represent a shift toward perception that is both biologically inspired and computationally efficient. This paradigm shift is expected to redefine the design of future vision systems, emphasizing continuous information flow, sparse computation, and adaptive intelligence [20, 22].

2.4.2 Applications in Autonomous Driving

Perception lies at the core of autonomous driving, enabling vehicles to understand and interact with their environment through continuous detection, segmentation, and tracking of surrounding entities such as cars, pedestrians, cyclists, and traffic signs. Conventional vision-based pipelines, often relying on frame-based cameras paired with deep networks like YOLO or SSD, have achieved substantial progress under controlled conditions. However, these systems face severe challenges in real-world driving: rapidly changing illumination between sunlight and shadows, nighttime scenes with high dynamic range, and high-speed motion that causes motion blur. Furthermore, high-resolution image streams require immense computational power and bandwidth, leading to elevated energy consumption and limited scalability on embedded automotive hardware [22, 23].

Event-based and neuromorphic vision systems directly address these challenges. Their microsecond-level temporal resolution enables the detection of fast-moving objects without motion blur, while their high dynamic range ensures reliable perception under both bright and low-light conditions. Moreover, their sparse and asynchronous output significantly reduces data transfer and computation demands, enabling real-time performance even on low-power processors. These properties are critical for automotive scenarios where timely reactions can prevent accidents and improve system efficiency [20, 17].

Recent research has demonstrated the growing potential of event-based object detection for autonomous vehicles. Approaches such as ASTMNet, Mixed-YOLO, and Recurrent Vision Transformers (RVT) process event streams to achieve robust detection under adverse weather and illumination. Other studies explore fusion-based architectures that combine event data with RGB frames to leverage the complementary strengths of both modalities. Large-scale datasets such as *MVSEC*, *DDD17*, *DSEC*, and the *Prophesee 1-Megapixel Automotive Dataset* provide real-world benchmarks for evaluating such models, containing synchronized event, frame, and inertial data recorded in dynamic driving scenarios [18, 20].

Within this context, neuromorphic algorithms based on SNNs extend these advantages by performing event-driven inference with minimal energy consumption. Networks incorporating biologically inspired mechanisms such as *short-term synaptic plasticity (STP)* can dynamically adapt to background activity and highlight motion patterns relevant to object detection. This adaptability enables the system to prioritize salient visual features—such as approaching obstacles or pedestrians—while suppressing irrelevant background noise. Consequently, the combination of event-based sensing and neuromorphic computation forms a unified perception framework capable of operating efficiently and robustly under the most demanding real-world conditions.

2.4.3 Related Work

This section reviews studies closely related to the proposed framework for event-based object detection and motion segmentation. Each work represents a significant contribution to neuromorphic vision but exhibits key limitations that motivate the present study. The discussion highlights how the proposed approach, based on short-term synaptic plasticity (STP) implemented through Tsodyks–Markram synapses, addresses these gaps.

Nagaraj et al. (2022) — DOTIE: Detecting Objects through Temporal Isolation of Events using a Spiking Architecture

Nagaraj et al. [25] introduced *DOTIE*, a lightweight spiking neural network designed for object detection using event cameras. Their approach isolates objects by separating event streams according to motion speed, leveraging the temporal structure of events to identify distinct motion patterns. The network consisted of a single-layer leaky integrate-and-fire (LIF) neuron model that grouped temporally similar events. After this temporal separation, spatial clustering was applied to localize object regions, resulting in a system that achieved low latency and energy efficiency without relying on supervised training or frame reconstruction.

Although innovative, DOTIE assumes that temporal isolation alone is sufficient for motion segmentation. It lacks adaptive synaptic dynamics, meaning all synapses respond identically to repeated stimuli, regardless of novelty or frequency. As a result, the model struggles in complex scenes with overlapping motions, background noise, or varying illumination. Additionally, it does not account for biologically realistic synaptic behaviors such as short-term depression or facilitation.

The proposed work extends DOTIE's principle by integrating *short-term synaptic plasticity* through the Tsodyks–Markram model. In the presented framework, depressing synapses suppress redundant background spikes, while facilitating ones enhance transient motion cues. This dynamic adjustment enables adaptive motion segmentation that is robust to noise and speed variations. Furthermore, the use of biologically detailed *iaf_psc_delta* neurons within the NEST simulator enhances temporal precision and interpretability compared to the simplified LIF neurons used in DOTIE.

Stoffregen et al. (2019) — Event-Based Motion Segmentation by Motion Compensation

Stoffregen et al. [26] proposed one of the earliest event-based motion segmentation algorithms, based on motion compensation. Their method aligned events by iteratively estimating motion parameters for each object or background region, using an expectation-maximization (EM)-like optimization process. The algorithm jointly optimized cluster membership and motion parameters, allowing per-event segmentation without explicit optical flow computation. This framework achieved robust segmentation under ego-motion and complex dynamic scenes.

However, the approach is computationally demanding and unsuitable for real-time operation on neuromorphic hardware. It relies on iterative parameter optimization and assumes motion can be modeled through simple geometric transformations such as translation or rotation. These assumptions break down for non-rigid or irregular object motion. Moreover, the algorithm lacks any biological foundation and does not exploit neural or synaptic computation principles.

In contrast, the proposed system performs segmentation through *biological temporal filtering* at the synaptic level. The Tsodyks–Markram synapses inherently modulate signal transmission based on recent activity, effectively separating dynamic and static regions in real time. This enables continuous, low-latency segmentation without the need for iterative motion estimation. The clustering module in the proposed work replaces the heavy EM optimization with a lightweight, spike-driven grouping mechanism for object localization.

Gehrig and Scaramuzza (2024) — Low-Latency Automotive Vision with Event Cameras

Gehrig and Scaramuzza [27] presented a hybrid event- and frame-based system for low-latency automotive perception. Their framework combined asynchronous event streams with conventional RGB frames, leveraging the high temporal resolution of events alongside the spatial detail of frames. This fusion achieved high detection performance at an effective rate of several thousand frames per second while maintaining competitive accuracy in autonomous driving benchmarks.

Despite its impressive performance, the hybrid design reintroduces frame-based sensing and deep-learning components, which compromises the fully neuromorphic paradigm. The system depends on supervised training, frame fusion, and high computational power, limiting its applicability in low-power or embedded contexts. It also lacks biological plausibility, as it does not incorporate spike-based computation or adaptive synaptic filtering.

The proposed framework remains fully *event-driven* and biologically inspired. By processing asynchronous event data directly through spiking neurons and STP-enabled synapses, it achieves temporal adaptability and low latency intrinsically, without requiring sensor fusion or deep-learning components. This makes it more energy-efficient and better aligned with the principles of neuromorphic computing.

Clerico et al. (2025) — Retina-Inspired Object Motion Segmentation for Event Cameras

Clerico et al. [28] developed a retina-inspired motion segmentation model based on *Object Motion Sensitivity (OMS)* circuits observed in the mammalian visual system. Their method emulated the center–surround receptive fields of bipolar and ganglion cells, using spatial convolutional filters to detect motion saliency while compensating for ego-motion. The model achieved parameter efficiency and effective motion segmentation by combining biological inspiration with computational practicality.

While biologically grounded, their approach aggregates events into short temporal frames to perform spatial convolutions, reintroducing frame-based latency and losing microsecond-level temporal precision. The model relies solely on spatial contrast computations and lacks mechanisms for temporal adaptation at the synaptic level. Consequently, it struggles with rapidly changing illumination or motion speed variability and does not exploit per-event processing.

The proposed work complements and extends this concept by introducing temporal adaptability through *short-term synaptic plasticity.* Instead of frame accumulation, each synapse dynamically modulates its efficacy on a per-event basis. This allows the system to preserve continuous temporal resolution and adapt in real time to scene dynamics. Functionally, the proposed STP mechanism serves as a temporal analogue to the retina's center–surround structure, enabling adaptive motion filtering entirely within the spiking domain.

Summary of Key Differences

In summary, previous works have demonstrated effective event-based motion segmentation through either temporal isolation, optimization-based modeling, or bio-inspired filtering. However, they often lack temporal adaptability, biological plausibility, or real-time efficiency. The proposed framework addresses these challenges by introducing *short-term synaptic plasticity* as a dynamic, biologically grounded mechanism for per-event temporal filtering. This approach enables continuous, energy-efficient, and adaptive motion segmentation, advancing the state of neuromorphic event-based perception.

Chapter 3

Methodology

3.1 Overview of the Proposed Framework

The objective of this work is to segment moving objects within event-based camera data by assigning bounding boxes to their spatial locations. In such recordings, each event represents a change in pixel intensity, which can be triggered by two main sources: the motion of objects in the scene and the ego-motion of the camera itself. When the camera is stationary, all events originate from moving objects, and no additional processing is required to isolate them. However, in most real-world scenarios, especially in automotive environments, the camera is in motion. This ego-motion produces a significant number of background events that do not correspond to meaningful object activity.

Because the speed and direction of ego-motion vary over time, a fixed filtering mechanism would fail to adapt to these changes. Therefore, the filtering process must dynamically respond to the temporal and spatial characteristics of the input. One key feature that the proposed framework exploits is locality, both in space and time. By leveraging this local structure, the system can distinguish between background noise caused by ego-motion and event patterns associated with independently moving objects.

The overall framework consists of two main stages: *filtering* and *clustering*. The filtering stage is implemented using the NEST simulator, which enables the modeling of biologically inspired spiking neural networks with various neuron and synapse configurations. Among the tested configurations, the most effective combines convolution-like spatial connectivity with depressing Tsodyks–Markram synapses, which naturally suppress repetitive background activity while enhancing transient motion features.

Once background events are filtered out, the remaining activity primarily corresponds to moving objects. The next stage applies a clustering algorithm to group these events into separate object regions. Because the number of objects in a scene is unknown a priori, density-based methods such as DBSCAN are well suited for this task. Besides isolating individual objects, clustering also helps remove residual noise by treating sparse or weakly correlated activity as outliers.

The following sections describe the methodology adopted in this study. First, the simulation environment based on the NEST simulator is presented, including details on the

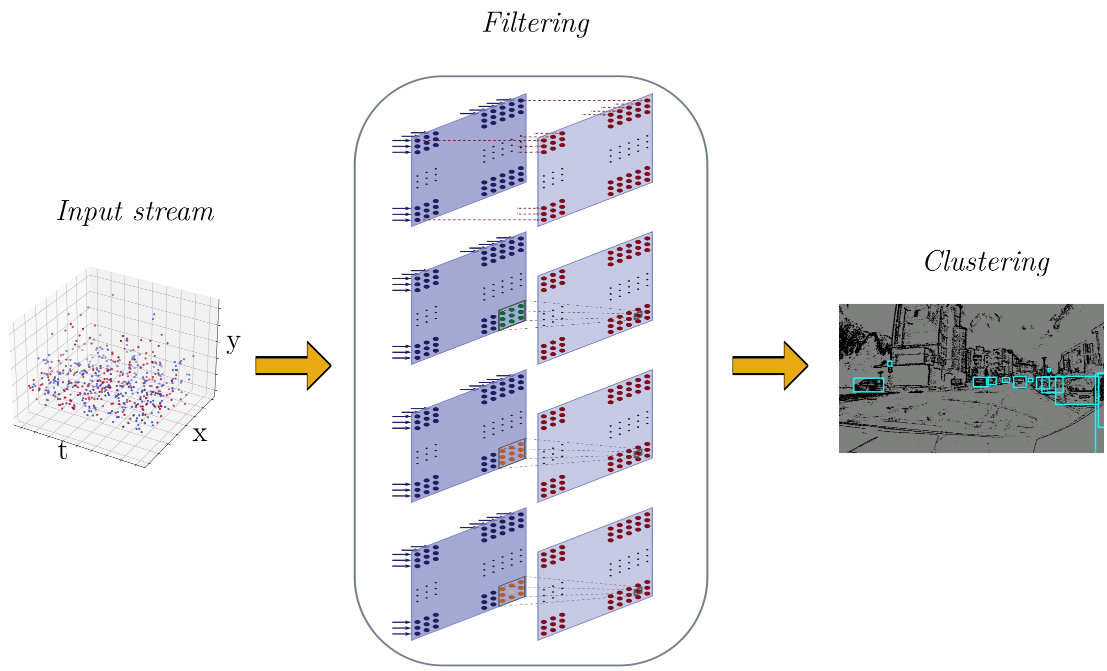


Figure 3.1: This figure provides an overview of the proposed framework. From left to right, the event-based input is first processed as either unipolar or bipolar depending on the selected filter configuration. The filtering stage then applies the spiking model to suppress background activity and emphasize motion related events. In the final stage, the clustering algorithm removes residual noise and groups the remaining events into object regions, where bounding boxes are generated.

neuron and synapse models used to construct the proposed filtering mechanisms. Next, the specific filtering configurations are introduced, illustrating how different neuron–synapse combinations can emphasize motion-related activity and suppress background noise. The subsequent section focuses on post-processing and clustering, where the filtered event data are grouped into object regions and bounding boxes are generated. Finally, the evaluation metrics used to assess the performance of the proposed framework are discussed.

3.2 NEST Simulation Environment

The NEST simulator is a high-performance simulation framework specifically developed for modeling large networks of spiking neurons. It provides an event-driven architecture optimized for the efficient propagation and scheduling of spike events, making it suitable for studying the dynamics of biologically inspired neural systems at different scales, from small microcircuits to large cortical networks.

In NEST, each neuron and synapse is represented as an individual process with its own state variables and update equations, while communication between elements occurs through discrete spike events transmitted along directed connections. The simulator employs a hybrid integration approach in which subthreshold dynamics are computed analytically between events. This ensures numerical precision, stability, and reproducibility across different simulation back ends and hardware configurations.

Beyond computational efficiency, NEST offers a modular and extensible structure that enables the definition of custom neuron, synapse, and plasticity models through its model library or user-defined extensions written in C++ or Python. This flexibility makes it a suitable platform for exploring biologically plausible mechanisms such as adaptation, recurrent connectivity, and short-term synaptic plasticity.

In this work, NEST serves as the core simulation environment for implementing the proposed spiking filters. The following sections describe the specific neuron and synapse models used in the simulations, along with their biophysical interpretations and computational formulations.

3.2.1 Neurons

iaf_psc_delta

The *integrate-and-fire* neuron with delta-shaped postsynaptic currents (`iaf_psc_delta`) is one of the fundamental neuron models implemented in the NEST simulator. It originates from the theoretical and numerical framework established by Rotter and Diesmann [29] and further utilized in large-scale network analyses by Diesmann et al. [30]. The model provides a minimal yet biophysically interpretable description of neuronal dynamics by separating subthreshold integration governed by a linear, time-invariant differential equation from nonlinear spike generation through threshold and reset mechanisms.

In this formulation, synaptic inputs are represented as Dirac delta functions, corresponding to instantaneous charge injections that cause discrete voltage jumps. This assumption allows the system to be treated within the framework of exact digital simulation, where the continuous subthreshold dynamics are analytically integrated between events. Despite its simplicity, this model captures essential features of neuronal computation such as temporal summation, refractoriness, and precise spike timing, while maintaining computational efficiency suitable for large-scale network simulations and state-space analyzes of synchronized spiking activity.

Parameter	Unit	Description
E_L	mV	Resting membrane potential
C_m	pF	Capacitance of the membrane
τ_m	ms	Membrane time constant
t_{ref}	ms	Duration of refractory period
V_{th}	mV	Spike threshold
V_{reset}	mV	Reset potential of the membrane
I_e	pA	Constant input current
V_{min}	mV	Absolute lower value for the membrane potential

Table 3.1: Parameters of the iaf_psc_delta model

The temporal evolution of the membrane potential V_m follows the differential equation:

$$\frac{dV_m}{dt} = -\frac{V_m - E_L}{\tau_m} + \dot{\Delta}_{syn} + \frac{I_{syn} + I_e}{C_m} \tag{3.1}$$

where E_L represents the resting potential, τ_m is the membrane time constant, $\dot{\Delta}_{syn}(t)$ denotes the rate of voltage change due to synaptic inputs, I_e is an external constant current, and C_m is the membrane capacitance.

A spike event occurs at time $t^* = t_{k+1}$ when the membrane potential crosses the threshold from below:

$$V_m(t_k) < V_{th} \quad \text{and} \quad V_m(t_{k+1}) \geq V_{th} \tag{3.2}$$

Following spike emission, the membrane potential is held at the reset value throughout the refractory interval:

$$V_m(t) = V_{reset} \quad \text{for} \quad t^* \leq t < t^* + t_{ref} \tag{3.3}$$

The contribution of synaptic inputs to membrane potential changes is expressed as:

$$\dot{\Delta}_{syn}(t) = \sum_j w_j \sum_k \delta(t - t_j^k - d_j) \tag{3.4}$$

where j indexes presynaptic neurons (with $w_j > 0$ for excitatory and $w_j < 0$ for inhibitory connections), k indexes individual spike times from neuron j, d_j represents the synaptic delay, and δ is the Dirac delta distribution. Each synaptic event produces an instantaneous voltage jump:

$$\Delta_{syn} = w \tag{3.5}$$

where w is the synaptic weight measured in millivolts. The corresponding postsynaptic current takes the form:

$$i_{syn}(t) = C_m \cdot w \cdot \delta(t) \tag{3.6}$$

resulting in a total charge transfer per synaptic event of:

$$q = \int_0^\infty i_{syn}(t)\, dt = C_m \cdot w \tag{3.7}$$

The model employs exact integration methods for subthreshold dynamics, ensuring numerical precision in membrane potential evolution. By default, the membrane potential is unbounded from below; however, a minimum voltage parameter V_{min} can be specified to prevent unphysical hyperpolarization. Synaptic inputs arriving during the refractory period are typically discarded.

3.2.2 Synapses

Static synapse

A static synapse is a synapse that does not exhibit any form of plasticity. It represents a straightforward connection, where the effective synaptic weight is simply the synaptic weight itself, as shown in Eq. 3.8.

$$x(t) \cdot w \tag{3.8}$$

Tsodyks Markram synapse

Parameter	Unit	Description
U	real	Parameter determining the increase in u with each spike [0,1]
τ_{psc}	ms	Time constant of synaptic current
τ_{fac}	ms	Time constant for facilitation
τ_{rec}	ms	Time constant for depression
x	real	Initial fraction of synaptic vesicles in the readily releasable pool [0,1]
y	real	Initial fraction of synaptic vesicles in the synaptic cleft [0,1]
u	real	Initial release probability of synaptic vesicles [0,1]

Table 3.2: Parameters of the tsodyks_synapse model

The TM synapse model describes a form of STP, where the efficacy of synaptic transmission depends on the recent history of presynaptic activity. Experimental observations show that, in certain neurons, previous presynaptic events modulate the probability of neurotransmitter release [31]. The TM model provides a mathematical formulation of this mechanism.

Two distinct scenarios are typically observed. In some synapses, frequent presynaptic activation leads to a depletion of available neurotransmitter resources, resulting in a gradual reduction of synaptic efficacy over time, a phenomenon known as Short-term Depression (STD). In contrast, in other synapses, repetitive presynaptic firing enhances synaptic strength due to calcium accumulation in the axon terminal, a process called Short-term Facilitation (STF).

These forms of short-term plasticity have been observed in various regions of the human nervous system, particularly within the visual cortex, where some areas are predominantly STD dominated while others exhibit stronger STF. Despite their seemingly opposite effects, both mechanisms can coexist within the same neural circuit. Importantly, since these processes do not involve permanent structural modifications, they represent transient, activity-dependent changes in synaptic efficacy.

Although the real dynamics are much more complex, a simplified formulation was proposed in the original paper. The synaptic effect evolves according to the following system of kinetic equations:

$$\frac{dx}{dt} = \frac{z}{\tau_{\text{rec}}} - u\,x\,\delta(t - t_{\text{sp}}), \tag{3.9}$$

$$\frac{dy}{dt} = -\frac{y}{\tau_{\text{I}}} + u\,x\,\delta(t - t_{\text{sp}}), \tag{3.10}$$

$$\frac{dz}{dt} = \frac{y}{\tau_{\text{I}}} - \frac{z}{\tau_{\text{rec}}}. \tag{3.11}$$

Here, x, y, and z represent the fractions of synaptic resources in the recovered, active, and inactive states, respectively. The variable t_{sp} denotes the timing of presynaptic spikes, τ_{I} is the decay constant of the Postsynaptic Currents (PSCs), and τ_{rec} is the recovery time constant from synaptic depression. Synaptic depression arises due to the depletion of vesicles in the readily releasable pool, corresponding to the variable x in Eq. 3.9.

Synaptic facilitation, on the other hand, results from an increase in the presynaptic release probability. This process is captured by the variable u, which evolves according to:

$$\frac{du}{dt} = -\frac{u}{\tau_{\text{facil}}} + U(1 - u)\;\delta(t - t_{\text{sp}}), \tag{3.12}$$

where τ_{facil} is the facilitation time constant, and U represents the baseline utilization of synaptic efficacy. When $\tau_{\text{facil}} \to 0$, the model reduces to a purely depressing synapse.

However, there are cases where this synapse model is applied to neuron models that do not include explicit PSCs dynamics. In such cases, the effective synaptic weight transmitted to the postsynaptic neuron upon the occurrence of a spike at time t is given by

$$u(t) \cdot x(t) \cdot w, \tag{3.13}$$

where $u(t)$ and $x(t)$ are defined in Eqs. 3.9 and 3.12, and w is the static synaptic weight specified during connection. This formulation can be interpreted in two components: the term $u(t).x(t)$ represents the probability of release times the amount of available synaptic resources, while w corresponds to the fixed synaptic efficacy.

The resulting product determines the amplitude of the synaptic impulse that triggers the postsynaptic response. The corresponding transmitter concentration $y(t)$ then decays back to zero with the time constant τ_{PSC}.

Depressing and Facilitating Behavior of TM Synapses

The Tsodyks Markram model captures both forms of STP through the interaction between two key variables: the release probability $u(t)$ and the fraction of available synaptic resources $x(t)$. Whether a synapse behaves as depressing or facilitating depends on how these variables evolve in response to repeated presynaptic spikes, and this evolution is controlled by the parameters U, τ_{rec}, and τ_{fac}.

A synapse exhibits *depressing* behavior when the baseline utilization U is high and the recovery time constant τ_{rec} is long. Under these conditions, repeated spikes cause the available resources $x(t)$ to decrease faster than they can replenish. With every spike, the product $u(t)x(t)$ becomes smaller, reducing the effective synaptic response. Depressing synapses therefore respond strongly to isolated or infrequent spikes but produce weaker responses during rapid or sustained presynaptic activity. In practical terms, this behavior suppresses high frequency inputs and can attenuate repetitive background activity, such as motion caused by camera ego movement.

In contrast, *facilitating* behavior emerges when the facilitation time constant τ_{fac} is large and the baseline utilization U is low. In this regime, repeated presynaptic spikes cause the release probability $u(t)$ to gradually increase. As $u(t)$ accumulates, the synapse becomes more effective with each spike. Facilitating synapses therefore produce modest responses to isolated spikes but increasingly strong responses to bursts or continuous presynaptic input. This mechanism enhances temporally consistent patterns and can emphasize features that persist within a localized region.

These two modes arise from the same mathematical model but represent opposite operating regimes. Depressing synapses reduce their contribution during sustained activity, whereas facilitating synapses strengthen their contribution. Both behaviors can coexist within the same neural circuit, and their relative influence can be controlled by adjusting the parameters. This flexibility allows the Tsodyks Markram model to reproduce a broad range of synaptic responses observed in biological systems.

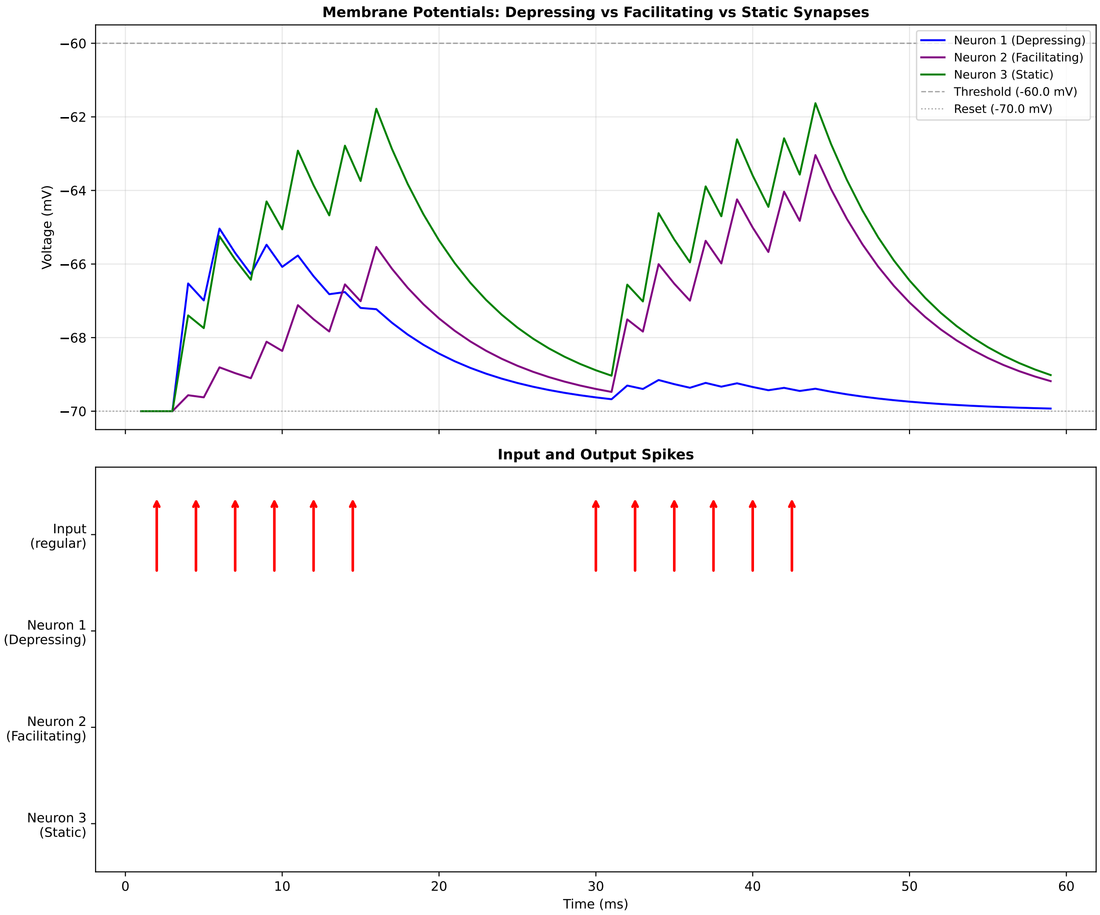


Figure 3.2: Comparison of synaptic plasticity effects with regular input pattern. Top: Membrane potentials of three neurons with depressing, facilitating, and static synapses. Bottom: Input spike pattern and corresponding output spikes from each neuron type.

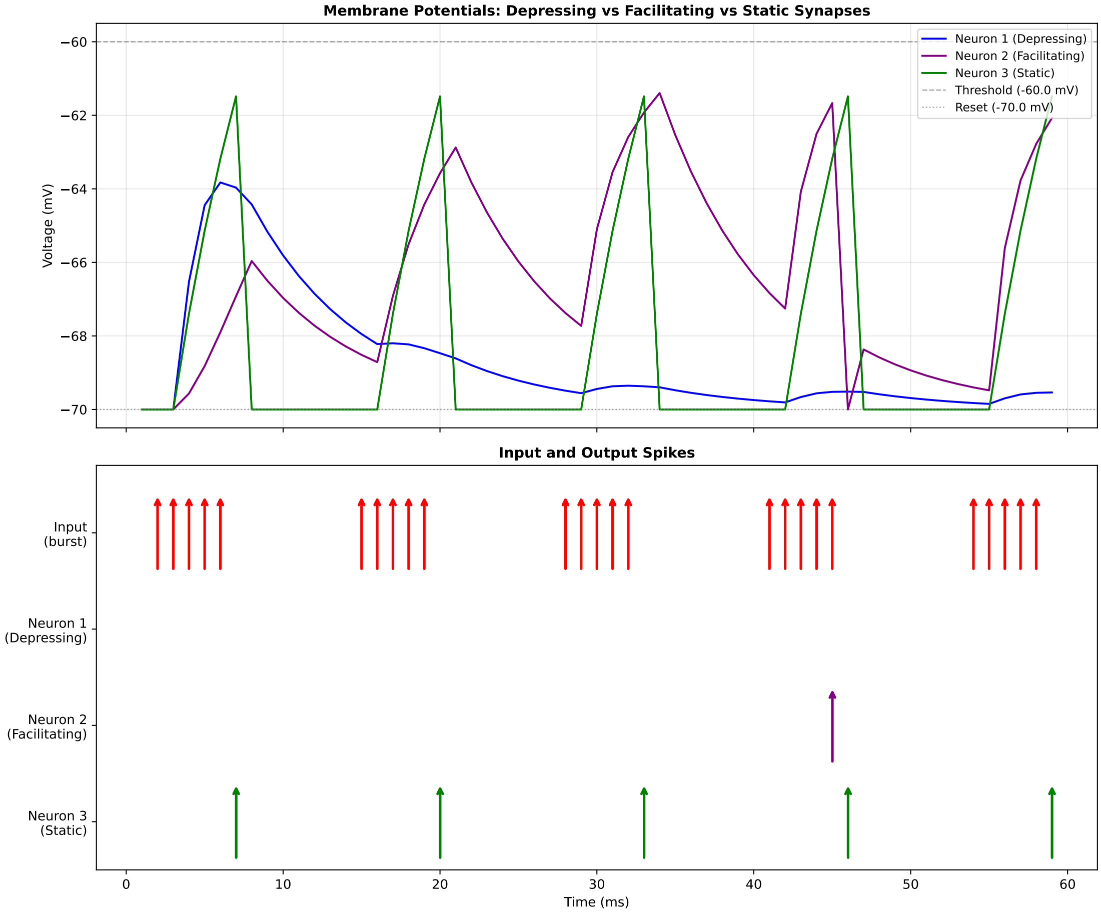


Figure 3.3: Comparison of synaptic plasticity effects with burst input pattern. Top: Membrane potentials of three neurons with depressing, facilitating, and static synapses. Bottom: Input spike pattern and corresponding output spikes from each neuron type.

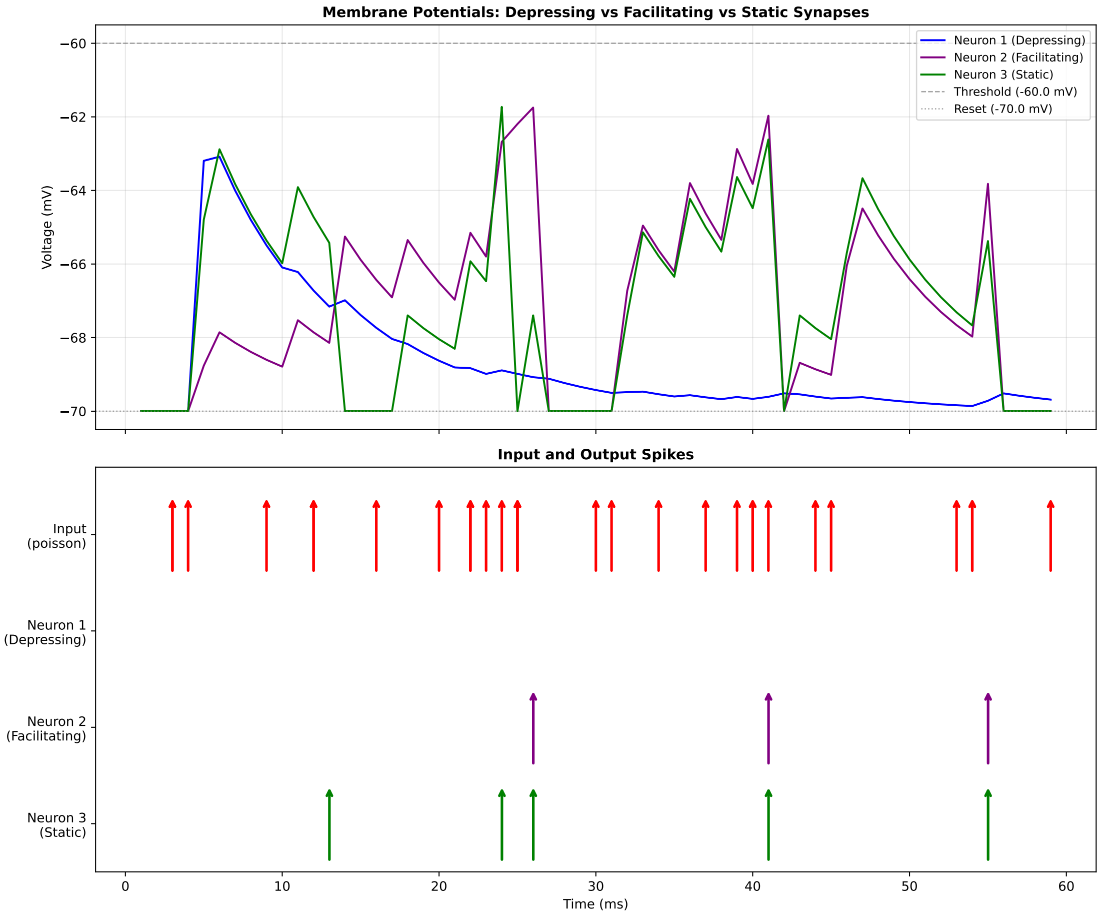


Figure 3.4: Comparison of synaptic plasticity effects with Poisson input pattern. Top: Membrane potentials of three neurons with depressing, facilitating, and static synapses. Bottom: Input spike pattern and corresponding output spikes from each neuron type.

3.3 Proposed Filtering Models

The filtering stage is responsible for suppressing background activity and highlighting event patterns that correspond to independently moving objects. Although the overall framework supports arbitrary neuron and synapse configurations, four specific filtering models were explored in this work. These models differ in two aspects: the spatial connectivity pattern between input events and filter neurons, and the synaptic mechanisms used to transmit spikes. Temporal processing is handled by the neuron dynamics in all configurations, while spatial processing is determined by how each model uses its local receptive field. The purpose of evaluating different configurations is to investigate how various combinations of locality and synaptic behavior influence event driven motion filtering.

3.3.1 Direct (One-to-One) with Static Synapses

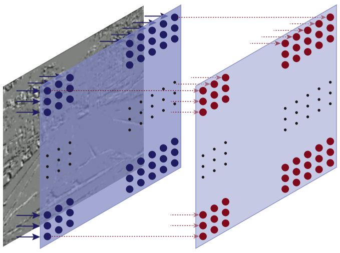
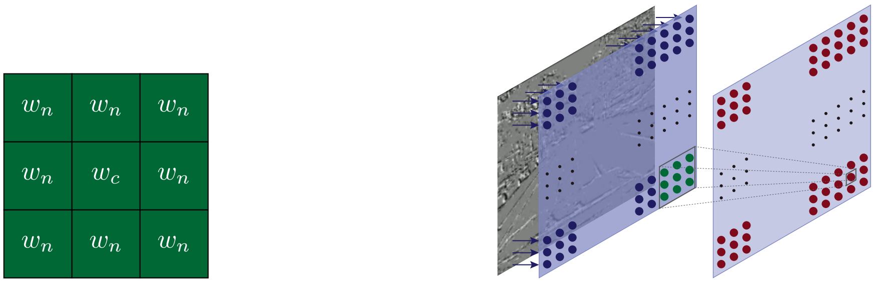


Figure 3.5: One to one connections.

In the first configuration, each input pixel is connected directly to a corresponding neuron in the filtering layer. All synapses are static and maintain a constant efficacy throughout the simulation. This setup does not incorporate any spatial interactions, since each neuron responds only to its own pixel without considering neighboring activity. The configuration serves as a structurally simple baseline that processes events independently across the spatial domain.

3.3.2 Convolution with Static Synapses

The second configuration introduces a convolution like connectivity pattern. Each neuron receives input from a small receptive field defined by a spatial kernel, allowing it to integrate information from its local neighborhood. The synapses remain static and do not change their strength over time. Typically, the central connection of the kernel is assigned a larger weight than the surrounding connections, providing stronger influence from the center pixel while still incorporating spatial context. This configuration mirrors the structure used in the DOTIE approach and allows the extraction of simple local motion cues [25].



w_n	w_n	w_n
w_n	w_c	w_n
w_n	w_n	w_n

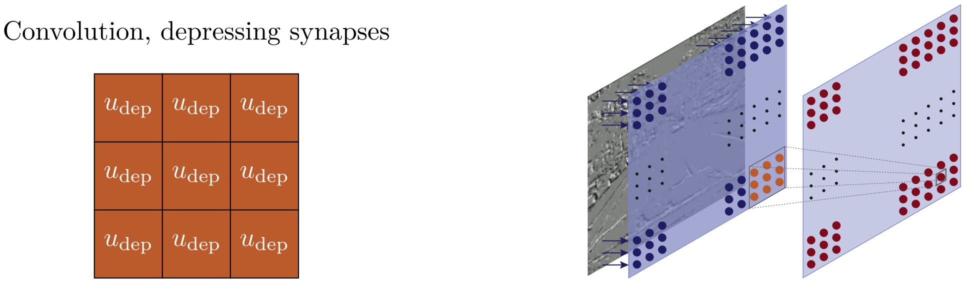
(a) Convolutional filtering configuration with static synapses.

(b) Spatial connections

Figure 3.6: Comparison of convolutional and spatial connectivity patterns.

3.3.3 Convolution with Depressing Synapses

In this configuration, the convolutional connectivity pattern is preserved, but all synapses in the kernel use depressing Tsodyks Markram dynamics. Depressing synapses temporarily reduce their efficacy in response to frequent activation, which enables the model to respond differently to repeated input patterns compared to isolated events. This configuration therefore incorporates both spatial locality and STP. The kernel is typically defined as a 3 × 3 receptive field, with the central connection usually assigned a higher weight than the surrounding ones.



Convolution, depressing synapses

u_{dep}	u_{dep}	u_{dep}
u_{dep}	u_{dep}	u_{dep}
u_{dep}	u_{dep}	u_{dep}

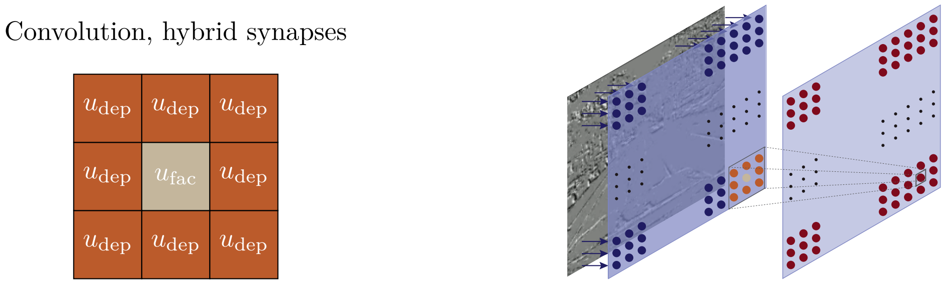
(a) Convolutional filtering configuration with depressing Tsodyks Markram synapses.

(b) Spatial connections with depressing synapses

Figure 3.7: Comparison of convolutional and spatial connectivity patterns under depressing synaptic dynamics.

3.3.4 Convolution with Hybrid Synapses

The hybrid configuration combines depressing and facilitating synapses within the same receptive field. The central connection employs a facilitating synapse whose efficacy increases when activated repeatedly, while the surrounding connections use depressing synapses. This design provides an asymmetric plasticity profile within the receptive field, allowing the model to explore how mixed short term synaptic dynamics interact with


Convolution, hybrid synapses
u_{dep} u_{dep} u_{dep}
u_{dep} u_{fac} u_{dep}
u_{dep} u_{dep} u_{dep}

(a) Hybrid convolutional filtering configuration with a facilitating center and depressing neighbours.

(b) Spatial connections with hybrid synapses

Figure 3.8: Comparison of hybrid convolutional connectivity and spatial hybrid connectivity patterns.

event distributions in dynamic scenes.

3.4 Post Processing and Clustering Algorithms

Although the spiking framework filters a large portion of background activity, the output still contains some residual noise and irrelevant events. In addition, multiple moving objects may appear in the scene simultaneously, and each must be identified as a separate entity according to the task definition. For this reason, the post processing stage consists of two main steps. First, the filtered event stream is converted into frame-like representations using a sliding window approach and then refined with morphological operations. Second, a clustering algorithm is applied to isolate individual objects and to generate bounding boxes around them.

3.4.1 Post Processing from Events to Frames

After the filtering stage, the output takes the form of an event cloud with coordinates in the x, y and t dimensions. The temporal resolution is determined by the simulation time step of NEST. In the experiments, a resolution of 1 millisecond was used to preserve as much temporal detail as possible while avoiding excessive computational overhead.

To apply morphological operations, the event cloud must first be converted into a frame. This is achieved with a sliding temporal window that accumulates events within a short time interval and projects them onto a two-dimensional grid. These frames are then processed with standard morphology functions such as dilation and opening, which help suppress noise and enhance the spatial continuity of object contours.

Because morphological operators may generate artificial pixels that did not exist in the original event stream, an additional logical AND operation is applied between the input frame and the processed frame. This step removes false positives introduced during filtering or morphology and ensures that only events supported by real activity remain in the final representation.

3.4.2 Clustering with DBSCAN

While various clustering algorithms can be applied to event-based data, they differ significantly in their assumptions and requirements. In the present task, the main challenge is that the number of moving objects is unknown at every moment of the sequence. This makes algorithms that require a predefined number of clusters unsuitable. Methods such as K-Means or Gaussian Mixture Models depend on specifying the number of clusters, and although heuristics like the silhouette score can be used to estimate this value, such strategies introduce considerable computational overhead and often produce unstable results over time, as also reported in the DOTIE framework [25].

In contrast, DBSCAN [32] is well suited for this scenario because it does not require the number of clusters to be specified. Instead, it identifies clusters as regions of higher point density that are separated by areas of lower density. This property aligns naturally with event-based motion segmentation, where each independently moving object forms a locally dense set of events, whereas background or residual noise remains comparatively sparse.

Mathematically, DBSCAN relies on two parameters: the neighborhood radius ε and the minimum number of points *MinPts*. A point is classified as a *core point* if its ε-neighborhood contains at least *MinPts* points. Clusters are then formed by iteratively expanding from these core points through the notions of *density reachability* and *density connectivity*. Points that satisfy the reachability condition are included in the cluster, border points are attached to the nearest dense region, and isolated points that do not meet any density requirement are labeled as noise. These definitions form the basis of the original DBSCAN formulation [32] and have been further analyzed and formalized in later studies [33].

A key advantage of DBSCAN in this application is its robustness to noise, which is inherent in event-based camera data. Even after the filtering stage, isolated events may appear due to sensor artifacts, weak reflections, or background inconsistencies. These events typically do not meet the density threshold and are therefore automatically discarded as noise. This behavior is highlighted in both the original DBSCAN paper and its subsequent theoretical analyses.

Another important advantage is DBSCAN's ability to detect clusters of arbitrary shape. Events generated by moving objects rarely form convex or regular patterns; instead, their shapes may be elongated, irregular, or fragmented depending on object geometry and motion. Algorithms that assume convex clusters struggle under such conditions. DBSCAN, being density based, adapts naturally to any geometry as long as local density is sufficient, which is particularly valuable for dynamic scenes with heterogeneous objects.

Finally, DBSCAN is computationally efficient for two dimensional spatial data, especially when neighborhoods remain small. The algorithm avoids iterative optimization and uses a deterministic expansion process governed entirely by the density parameters. This makes DBSCAN a practical choice for integration into an event-based detection pipeline, where consistent performance and low overhead are essential.

3.5 Evaluation Metrics

Evaluating event-based object segmentation requires comparing the predicted bounding boxes with the available ground truth annotations in each dataset. Some datasets, such as the 1 Megapixel Automotive Detection dataset, provide ground truth bounding boxes directly. Others, such as MVSEC, do not include bounding box annotations, and ground truth information must be generated externally. Following common practice, ground truth boxes for MVSEC were produced by applying a conventional frame based object detector such as YOLOv3 to the grayscale camera stream.

Even in datasets that already contain ground truth, these annotations originate from frame based models that operate at a much lower frame rate than event-based systems. Consequently, there is a temporal mismatch between the annotations and the proposed framework, which produces predictions at every time window. For this reason, evaluation is performed only at the timestamps for which ground truth bounding boxes are available. Predictions at intermediate timestamps are not considered in the scoring process.

For each time window with ground truth, the predicted bounding boxes are compared with the ground truth using the Intersection over Union (IoU) metric, defined as

$$\mathrm{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}. \tag{3.14}$$

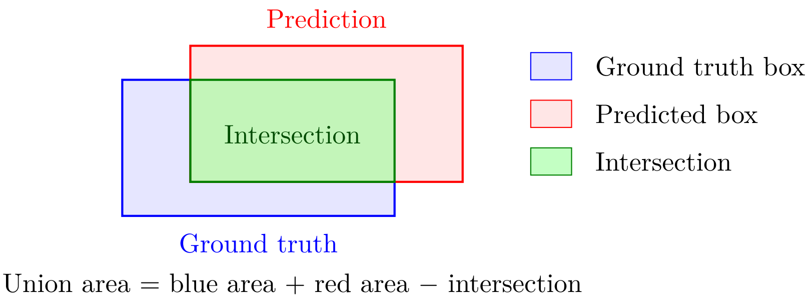


Figure 3.9: Illustration of Intersection over Union (IoU) between a ground truth bounding box and a predicted bounding box. IoU is defined as the ratio between the area of the intersection and the area of the union of the two boxes.

A predicted box is counted as a True Positive (TP) if its IoU with a ground truth box exceeds a threshold of 0.5. When IoU < 0.5, the spatial overlap is considered insufficient and the prediction is treated as a False Negative (FN). This threshold is motivated by the fact that an IoU value below 0.5 indicates that more than half of the predicted or ground truth region does not overlap, implying that the detection is too inaccurate to be considered correct. If a predicted box does not overlap with any ground truth box, it is labeled as a False Positive (FP).

Based on the counts of TP, FP and FN, standard evaluation metrics are computed. Precision measures the proportion of correctly predicted objects, recall quantifies how many of the actual objects were detected, and the F1 score provides a balanced combination of the two. Their definitions are

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{3.15}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{3.16}$$

$$\text{F1}\,\text{score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3.17}$$

These metrics collectively evaluate the correctness and completeness of the detections, providing a reliable and widely used means of assessing the performance of object detection systems, including event-based approaches such as the one proposed in this work.

Chapter 4

Results & Experiments

4.1 Experimental setup

4.1.1 Datasets

Prophesee 1 Megapixel Automotive Detection Dataset

The Prophesee 1 Megapixel Automotive Dataset [34] is the first large-scale, high-resolution dataset dedicated to event-based object detection in real driving conditions. It was designed to bridge the gap between low-resolution neuromorphic recordings and frame-based automotive benchmarks by combining long-duration event sequences with dense object-level annotations.

The data were collected using a 1280×720 event camera mounted behind the windshield of a car, alongside an RGB camera. Both cameras were fixed on a rigid mount with minimal baseline distance to reduce parallax. The RGB stream was recorded at 4 megapixels and 60 Hz, ensuring temporal and spatial overlap with the asynchronous event stream.

A fully automated labeling protocol was developed to generate bounding-box annotations for the event data. The process consisted of three main steps:

1. Temporal synchronization of the RGB and event cameras, achieved through either a physical trigger or an algorithmic cross-correlation procedure;

2. Detection of objects in each RGB frame using a high-performance commercial automotive detector that identified cars, pedestrians, and two-wheelers;

3. Spatial mapping of the detected bounding boxes from the RGB image plane to the event-camera coordinates using a homography transformation.

This automatic transfer method produced temporally aligned and geometrically consistent labels at a frequency of 60 Hz. The synchronization and homography assumptions introduced only minor deviations which remained negligible in the context of typical road-scene distances.

The final dataset comprises approximately 14.65 hours of recordings, divided into 11.19 hours for training, 2.21 hours for validation, and 2.25 hours for testing. Recordings cover a broad range of environments including dense urban traffic, highways, suburban neighborhoods, and rural roads, collected over several months and under diverse daylight and weather conditions. The resulting corpus contains more than 25 million bounding boxes, providing dense spatiotemporal supervision at automotive scale.

For this work, the availability of precise, high-frequency bounding boxes allows the evaluation of event-driven detection and tracking approaches without relying on frame reconstruction. Owing to its high spatial resolution, temporal density, and automated annotation pipeline, the Prophesee 1 Megapixel Automotive Dataset serves as a robust benchmark for assessing event-based object-detection algorithms in challenging real-world conditions.

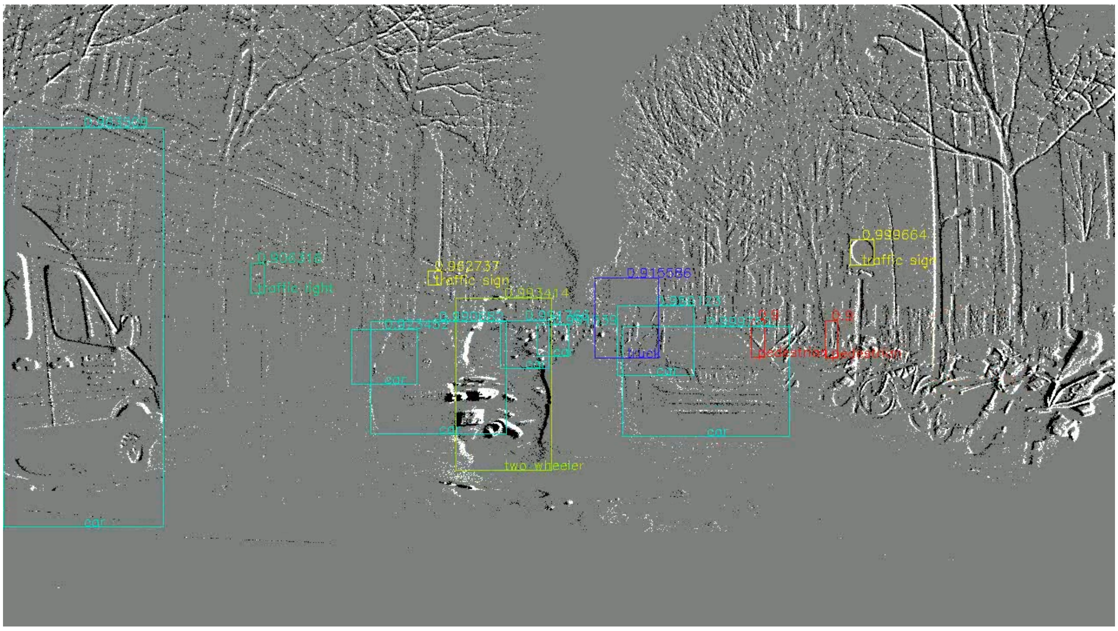


Figure 4.1: Example scene from Prophesee 1 Megapixel Automotive Dataset. This frame is generated by accumulating events in 1/60 second of a time window

MVSEC Dataset

The MVSEC Dataset [35] provides multimodal recordings collected using a stereo pair of event-based cameras, grayscale cameras, Inertial Measurement Unit (IMU) sensors, and a Light Detection and Ranging (LiDAR). The data was recorded on multiple platforms including a handheld rig, a hexacopter, a motorcycle, and a car.In this thesis, only the car-mounted sequences are used, as they provide large-scale, outdoor, and dynamic driving scenarios that closely resemble real-world traffic environments.

In the car configuration, the stereo cameras were installed on the sunroof of a sedan, facing forward with a slight downward pitch to capture the road and surrounding traffic. The sequences include both daytime and evening recordings, with vehicle speeds reaching up to 12 m/s. Each DAVIS camera captures both asynchronous events and grayscale frames at a resolution of 346 × 260 pixels. The grayscale frames serve as low-rate reference images and are timestamp-aligned with the event stream.

Since the MVSEC dataset itself does not include bounding-box annotations, this work leverages the DOTIE framework [25] to extract bounding boxes for moving objects from the grayscale frames. In its original formulation, DOTIE uses YOLOv3 detections on MVSEC grayscale frames as ground truth to validate its event-based segmentation results.

In this thesis, the same DOTIE pipeline is applied to generate bounding boxes aligned with the grayscale frames of the MVSEC car-day and car-evening sequences. These bounding boxes define spatial regions corresponding to moving objects (primarily vehicles) and are used as region-of-interest references for evaluating event-based detection performance. The resulting annotations preserve the asynchronous nature of the event data while maintaining geometric alignment with the corresponding grayscale imagery.

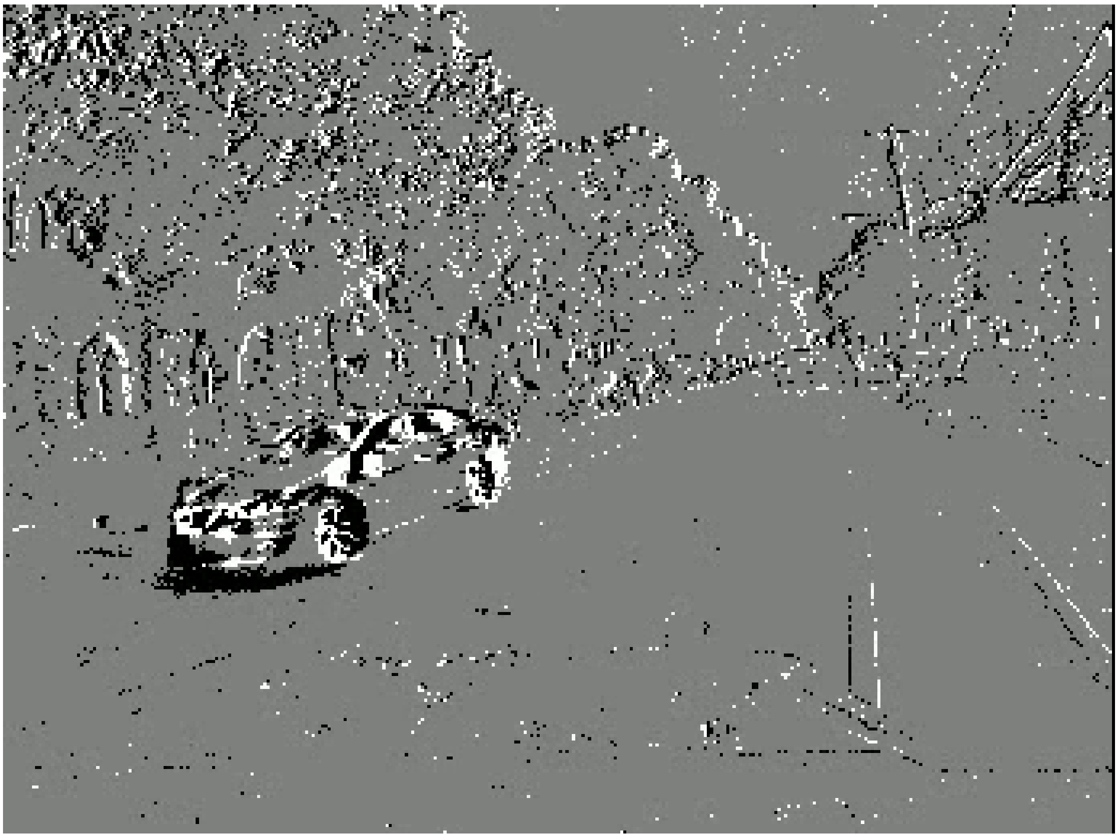


Figure 4.2: Example scene from MVSEC. This frame is generated by accumulating events in 1/30 second of a time window

4.1.2 Experiment Configuration

The experiments are conducted using both datasets introduced in the previous section. Due to the large size and computational demands of these recordings, only representative portions are selected for simulation. For the MVSEC dataset, the *outdoor_day_2 sequence* is used, which consists of a single recording of approximately eleven minutes. For the Prophesee dataset, the *validation_2* split is selected, containing twenty nine recordings of one minute each. Since running NEST simulations on the full duration of these videos is not practical, each sequence is divided into shorter temporal segments. MVSEC recordings are partitioned into twenty second windows, while the Prophesee videos are segmented into ten second windows. Each segment is then processed independently to allow parallelism and reduce memory requirements.

As explained earlier, four groups of filtering models are evaluated. These include direct one to one connections, spatial filtering kernels, depressing synapses based on STD, and hybrid models that combine facilitation and depression. Within each model group, multiple membrane time constants are tested to investigate the influence of temporal dynamics on event integration. For plastic synapse models, different strengths of STD or STF are also explored to understand how synaptic adaptation affects the suppression of background activity and the highlighting of moving targets.

After filtering, a clustering stage groups the remaining spike activity into spatially coherent detections. This step is essential because the filtering models output binary spike maps rather than explicit object regions. A combination of spatial recovery, density based clustering, and size pruning is employed to convert these spike patterns into bounding box proposals. Multiple parameter sets are tested to evaluate the stability of the clustering pipeline across scenarios that vary in object density, motion, and noise levels.

Table 4.1: Summary of all NEST simulation configurations used in the experiments. The table is organized by model complexity, from simple static connections to hybrid STP mechanisms.

Model	Dynamics	Kernel	Synapse	Notes
Model 1 (Direct)	Fast	1×1	Static	No spatial interaction, no plasticity
	Moderate fast	1×1	Static	
	Slow	1×1	Static	
	Slowest	1×1	Static	
Model 2 (Only Spatial Filter)	Fast	3×3	Static	Spatial kernel, no plasticity
	Slow	3×3	Static	
	Moderate fast	3×3	Static	
	Slowest	3×3	Static	
Model 3 (Depression)	Fast	3×3	TM	Spatial kernel with weak or strong STD
	Slow	3×3	TM	
	Fast	3×3	TM	
	Slow	3×3	TM	
Model 4 (Hybrid)	Fast	3×3	TM	Spatial kernel with weak or strong STD and STF
	Slow	3×3	TM	
	Fast	3×3	TM	
	Slow	3×3	TM	

Table 4.2: Clustering and post processing parameters explored in the grid search.

Parameter	**Values**	**Explanation**
`recovery_neighborhood`	5, 7, 10, 12, 15	Controls the spatial extent used to reconnect fragmented activity regions. Larger values increase region continuity but may merge nearby objects.
`eps_val`	5, 7, 10, 12, 15	Distance threshold for density based clustering. Larger values allow broader clusters, while smaller values produce more compact ones.
`min_samples_val`	5, 7, 10, 12, 15	Minimum number of events required to form a valid cluster. Higher values suppress noise but may remove small or weak objects.
`mindiagonalsquared`	2000, 2300	Minimum bounding box diagonal squared. Filters out very small detections and enforces a lower size limit.

4.2 Results

This section presents the quantitative results obtained from the event-based filtering and clustering pipeline described in the previous chapters. Four filtering approaches are evaluated on two datasets using a combination of temporal dynamics and clustering parameters. The tables report, for each filter type, the configuration that achieved the best overall balance across precision, recall, F1 score, and mean Intersection over Union (IoU). Unless stated otherwise, the IoU threshold used for matching detections is 0.5.

The results for each dataset are presented in pairs, with the standard threshold table followed by the corresponding reduced threshold table. This organisation allows direct comparison of the performance under the default IoU requirement and a relaxed threshold of 0.25, which provides an additional perspective on filters that tend to generate detections with partial but consistent spatial overlap.

Prophesee 1 Megapixel Dataset

Results at IoU Threshold 0.5

Table 4.3 reports the best performing configuration from each filtering family under an IoU threshold of 0.5. The direct and spatial filters achieve the highest overall F1 scores, both reaching 0.157. Their precision and recall values are also relatively balanced. The direct filter attains a precision of 0.169 and a recall of 0.147, while the spatial filter yields a precision of 0.158 and a recall of 0.155. Both models produce a large number of false positives, with the direct filter generating 89510 false detections and the spatial filter generating 98976.

In contrast, the depressing and hybrid filters produce substantially lower F1 scores, both around 0.027. Their precision remains low, with values of 0.020 for the depressing model and 0.021 for the hybrid model. Recall follows a similar trend, remaining limited at 0.041 and 0.042 respectively. These models generate fewer true positives compared to the static filters, with 391 detections for the depressing model and 743 for the hybrid model. The number of false positives is also considerably lower: 18888 for the depressing filter and 34741 for the hybrid filter. Despite the reduced activity, both filters preserve mean IoU values above 0.60, which is comparable to the static filters.

Overall, the depressing and hybrid filters operate with reduced detection counts and lower recall, accompanied by significantly fewer false positives, while the direct and spatial filters produce higher activity levels, higher recall, and substantially larger FP values.

Filter	**Dynamics**	**Precision**	**Recall**	**F1**	**Mean IoU**	**TP**	**FP**	**FN**
Direct	Slowest	0.169	0.147	0.157	0.657	18254	89510	106081
Spatial	Moderate Fast	0.158	0.155	0.157	0.648	18522	98976	100646
Depressing	Slow	0.020	0.041	0.027	0.610	391	18888	9125
Hybrid	Slow	0.021	0.042	0.0279	0.605	743	34741	17118

Table 4.3: Top performing filtering configurations for the Prophesee 1 Megapixel dataset (IoU threshold 0.5).

Results at IoU Threshold 0.25

Lowering the IoU threshold from 0.5 to 0.25 results in substantial metric increases for both depressing and hybrid filters. For the depressing filter, precision increases from 0.020 to 0.098, recall rises from 0.041 to 0.218, and the F1 score increases from 0.027 to 0.135. The number of true positives grows from 391 to 2125, while false positives remain similar, increasing only from 18888 to 19486. False negatives decrease from 9125 to 7584. The mean IoU decreases from 0.610 to 0.396.

A similar pattern is observed for the hybrid filter. Precision improves from 0.021 to 0.095, recall increases from 0.042 to 0.199, and the F1 score increases from 0.0279 to 0.128. True positives more than double, from 743 to 1668. False positives remain lower than those of the static filters, increasing moderately from 34741 to 15958. False negatives drop from 17118 to 6723. As with the depressing filter, the mean IoU decreases from 0.605 to 0.390.

When compared with the direct and spatial filters at the 0.5 threshold, the reduced threshold depressing and hybrid filters remain below their F1 scores of 0.157. Precision values of both depressing and hybrid filters move closer to the ranges of the direct and spatial filters, which lie between 0.158 and 0.169. Their false positive counts remain much lower than those of the static filters, which exceed 89500 for the direct model and 98900 for the spatial model. This yields a substantially different balance between true and false detections, although the mean IoU values remain lower than those of the static filters.

Filter	Dynamics	Precision	Recall	F1	Mean IoU	TP	FP	FN
Depressing	Slow	0.098	0.218	0.135	0.396	2125	19486	7584
Hybrid	Fast	0.095	0.199	0.128	0.390	1668	15958	6723

Table 4.4: Best performing depressing and hybrid filters for the Prophesee 1 Megapixel dataset under a reduced IoU threshold of 0.25.

MVSEC Dataset

Results at IoU Threshold 0.5

Table 4.5 reports the best performing configurations for each filter type using an IoU threshold of 0.5. Among all models, the depressing filter achieves the highest F1 score with a value of 0.090, followed by the hybrid filter at 0.086. Their precision values are 0.074 and 0.071 respectively, which exceed those of both the direct and spatial filters. Recall values for the depressing and hybrid models are 0.118 and 0.109.

The direct filter attains a precision of 0.054 and a recall of 0.163, resulting in an F1 score of 0.081. It generates the highest number of true positives, with 1133 detections, but also produces the largest number of false positives at 19786. The spatial filter shows a precision of 0.045 and a recall of 0.142, with an F1 score of 0.068. It produces 137 true positives and 2876 false positives.

Compared to the static filters, both depressing and hybrid models exhibit higher precision and higher F1 scores, despite generating fewer detections overall. The depressing filter yields 51 true positives and 638 false positives, while the hybrid filter reports 30 true positives and 395 false positives. The false negative counts for the depressing and hybrid filters are 383 and 245 respectively. Their mean IoU values, 0.638 for the depressing model and 0.607 for the hybrid model, are comparable to those of the static filters.

Overall, the depressing and hybrid filters achieve the highest precision and highest F1 values among all four filter types, while the direct and spatial filters produce higher recall and substantially larger false positive counts.

Filter	Dynamics	Precision	Recall	F1	Mean IoU	TP	FP	FN
Direct	Slow	0.054	0.163	0.081	0.639	1133	19786	5823
Spatial	Slowest	0.045	0.142	0.068	0.633	137	2876	827
Depressing	Slow	0.074	0.118	0.090	0.638	51	638	383
Hybrid	Slow	0.071	0.109	0.086	0.607	30	395	245

Table 4.5: Top performing filtering configurations for the MVSEC dataset (IoU threshold 0.5).

Results at IoU Threshold 0.25

Lowering the IoU threshold from 0.5 to 0.25 leads to clear increases in all core detection metrics for both depressing and hybrid filters. For the depressing filter, precision increases from 0.074 to 0.146, recall rises from 0.118 to 0.266, and the F1 score increases from 0.090

to 0.188. The number of true positives grows from 51 to 132, while false positives increase moderately from 638 to 772. False negatives decrease from 383 to 365. The mean IoU decreases from 0.638 to 0.468.

The hybrid filter shows similar improvements. Precision increases from 0.071 to 0.160, recall increases from 0.109 to 0.273, and the F1 score increases from 0.086 to 0.201. True positives more than double, rising from 30 to 75. False positives remain low, at 395 in this configuration, and false negatives decrease from 245 to 200. The mean IoU decreases from 0.607 to 0.457.

When compared to their counterparts at the standard threshold, both filters achieve substantially higher precision, recall, and F1 scores under the reduced threshold while maintaining relatively low false positive counts. Their gains in recall are especially pronounced, increasing by more than a factor of two for both models.

Compared to the direct and spatial filters at IoU 0.5, the reduced-threshold depressing and hybrid filters exceed all static models in both precision and F1 score. The hybrid filter reaches an F1 value of 0.201, surpassing the direct filter's 0.081 and the spatial filter's 0.068. Their precision values, 0.146 and 0.160, are also considerably higher than the precision of the direct and spatial models, which remain at 0.054 and 0.045. Although their recall values stay below the direct filter's 0.163 and the spatial filter's 0.142, the reduced-threshold depressing and hybrid filters maintain significantly lower false positive counts, remaining far below the 19786 and 2876 false positives produced by the direct and spatial filters respectively.

Overall, under the reduced IoU threshold, the depressing and hybrid filters achieve the highest precision and highest F1 scores among all filtering models, while still producing substantially fewer false positives than the static filters.

Filter	**Dynamics**	**Precision**	**Recall**	**F1**	**Mean IoU**	**TP**	**FP**	**FN**
Depressing	Slow	0.146	0.266	0.188	0.468	132	772	365
Hybrid	Slow	0.160	0.273	0.201	0.457	75	395	200

Table 4.6: Best performing depressing and hybrid filters for the MVSEC dataset under a reduced IoU threshold of 0.25.

4.3 Discussion

The results presented in the previous section highlight how different filtering strategies behave under the constraints of real-world event camera datasets and model-generated ground truth. While the quantitative metrics provide an initial indication of performance, their interpretation requires careful consideration of the datasets, the annotation process, and the operational goals of the proposed filtering approach. This discussion examines these factors in detail, focusing on the interaction between dataset characteristics and evaluation metrics, the behavior of the filtering models relative to their intended function, and the influence of the clustering pipeline on the final detections. Together, these analyses provide a clearer understanding of the strengths and limitations of the proposed method in its current form.

4.3.1 Dataset and Metrics

This work evaluates an event-based filtering and segmentation algorithm using two datasets that were not originally designed for pixel level segmentation in the event domain. Both the Prophesee 1 Megapixel and MVSEC datasets provide ground truth in the form of bounding boxes extracted by applying an object detection model to RGB or grayscale video streams. The event data themselves do not include explicit per event segmentation labels. As a result, the evaluation pipeline is constrained by the limitations of frame based annotations transferred to an asynchronous domain.

Although the algorithm is fundamentally designed for segmentation of event streams, it can be adapted to provide bounding box detections through post processing, as demonstrated in this thesis. However, this is not an ideal evaluation setting. Attempts to obtain segmentation ground truth directly from the event data using state-of-the-art models such as SAM 2 [36] produced poor results, particularly because only the MVSEC dataset contains grayscale frames and because the temporal density of events far exceeds what current segmentation models are designed to process. Furthermore, the Prophesee dataset does not provide RGB footage at all. Even when non event-based segmentation can be estimated, the inherent frame rate mismatch poses a fundamental constraint: typical RGB or grayscale cameras operate at around 60 Hz, whereas event cameras produce updates at microsecond resolution with effective rates approaching 100000 Hz. Consequently, only event windows that correspond to the timestamps of the available ground truth can be evaluated.

A more significant challenge arises from the nature of the ground truth itself. The bounding box annotations in both datasets identify all objects present in the scene, regardless of whether they are moving or static. In contrast, the filtering algorithm is explicitly designed to highlight moving objects and suppress static background structures. This mismatch between annotation semantics and algorithmic intent complicates the interpretation of standard detection metrics.

The DOTIE evaluation strategy partially addresses this by defining a true positive when the predicted box reaches an IoU of at least 0.5 with a ground truth box. If the IoU falls below this threshold, the prediction is counted as a false negative, while predictions with no matching ground truth are considered false positives. This procedure implicitly assumes that every moving object is captured sufficiently well in at least one ground truth bounding box. In practice, this assumption is often invalid. For example, a moving object may be detected by the filter but represented in the ground truth with low spatial accuracy due to limitations of the frame based detector. The resulting low IoU leads to a false negative assignment despite the algorithm correctly identifying the motion.

Conversely, static objects that appear in the ground truth are sometimes erroneously highlighted by the filtering process because of noise or ego-motion. In these cases, the evaluation protocol counts them as true positives or false negatives based on IoU, even though they should be labeled as false positives from the perspective of motion based detection. These inconsistencies explain why the total number of true positives and false negatives varies across experiments despite representing the same scenes. Since the ground truth is itself model generated rather than manually annotated for motion, standard metrics such as precision, recall, and F1 score become difficult to interpret reliably.

Based on these observations, the most informative metric in this context is the number of false positives. False positives directly measure how effectively a filter suppresses background activity, ego-motion artifacts, and noise driven spiking. For a motion based segmentation algorithm, a low false positive rate provides strong evidence that the filter is performing its intended role even when precision and recall are distorted by the limitations of the ground truth annotation process.

4.3.2 Filters

The behavior of the four filtering configurations illustrates how spatial connectivity, temporal integration, and synaptic dynamics shape the ability of the system to extract motion related event patterns.

Direct Filter The direct one to one configuration operates without any spatial integration and relies entirely on the temporal characteristics of individual neurons. Each neuron receives input from a single pixel, allowing it to function as an independent temporal filter. This setup is effective for attenuating transient noise and high frequency artifacts caused by sensor distortion, since the neuronal decay parameter naturally suppresses isolated spikes that are not part of coherent motion. The same mechanism also enables implicit speed selectivity, because event streams with different temporal densities interact differently with the membrane potential dynamics. As a result, motion components outside the time constant of the neuron decay rapidly and contribute little to the output. However, the absence of spatial context means that this configuration cannot suppress noise that manifests across multiple neighboring pixels, nor can it group related events into larger spatial structures.

Spatial Filter The spatial filtering configuration extends the direct model by incorporating a convolution like receptive field. Each output neuron integrates events from a local neighborhood, which enables suppression of isolated background activity that does not match the pattern supported by the kernel. This spatial coupling is particularly effective when the scene contains limited ego-motion, since local neighborhoods remain relatively stable and background noise tends to occur as scattered spurious activations. In static or low motion conditions, the kernel structure therefore helps the model emphasize coherent activity while reducing random firing.

However, the effectiveness of the spatial filter diminishes significantly in the presence of strong ego-motion. Large scale motion causes widespread activation across many neighboring pixels, which appears coherent to the convolutional kernel even when it originates from the background. As a result, the receptive field amplifies rather than suppresses ego-motion artifacts, making it difficult for the filter to distinguish background motion from independent object motion. Varying the kernel size can shift the balance between noise suppression and spatial smoothing, but the fundamental limitation remains the same.

Depressing Filter The depressing configuration introduces STP into the filtering process. Each synapse reduces its efficacy in response to repeated activation, which allows

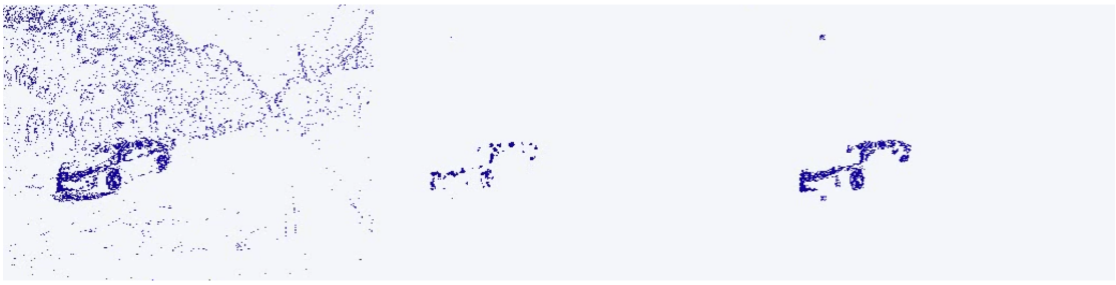


Figure 4.3: Spatial filter works well when background noise caused by ego-motion is not strong. Left is input, middle is the raw output after filter, right is post processed output

the filter to adapt to persistent background motion. This adaptive mechanism is highly effective at suppressing noise patterns generated by ego-motion, since large static structures such as buildings or the road surface produce dense, repeated event activity as the camera moves. Over time, the synapses corresponding to these regions become strongly depressed and largely cease to transmit spikes.

Because the model preserves the convolutional connectivity pattern, it retains both spatial and temporal processing capabilities. The receptive field allows the filter to respond to coherent spatial structure within moving objects, while synaptic depression modulates responses based on temporal frequency. However, the behavior of the filter is sensitive to the adaptation speed of the synapses. Incorrect settings can lead either to insufficient background suppression or to excessive depression that removes relevant motion cues.

A more fundamental limitation appears when large static structures dominate the scene. If a building or wall generates sustained event activity due to ego-motion, the synapses in that region rapidly depress and remain inactive. When a moving object later enters the same region, the filter may fail to detect it because the synapses have not recovered. A similar effect occurs inside moving objects: as an object generates repeated activations, its interior synapses gradually depress, causing the filter to detect mainly the leading edge. This reduces the spatial extent of detections and often lowers IoU, which motivates evaluating these models with reduced IoU thresholds.

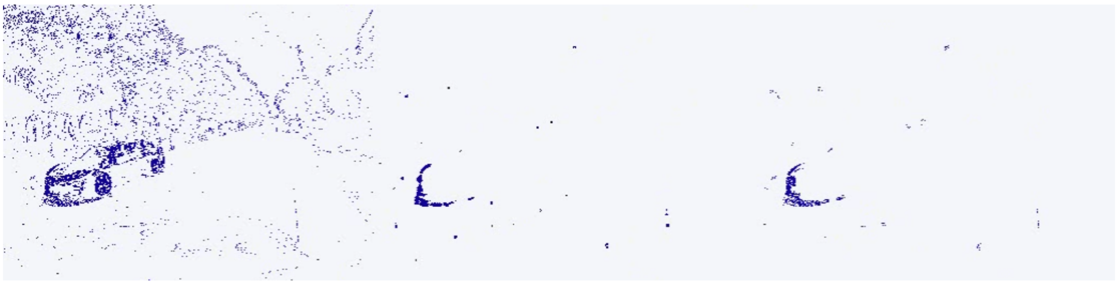


Figure 4.4: Depressing filter is good at surpassing the noise, struggling the detect rest of the object. Left is input, middle is the raw output after filter, right is post processed output

Hybrid Filter The hybrid configuration combines a facilitating synapse at the center of the receptive field with depressing synapses in the surround. In principle, this design aims to preserve central object activity while suppressing background motion. In practice, facilitation does not accumulate quickly enough to be effective. Moving objects often do not generate the rapid repeated activations required to drive facilitation to useful levels.

Facilitation can also interact negatively with background events. After an object leaves the region, the facilitating synapse may strengthen in response to background activity, producing unwanted traces behind moving objects. Additionally, placing the facilitating synapse directly at the input makes it sensitive to random background spikes, causing it to remain active long after the object has passed. These effects limit the practical advantage of adding facilitation and introduce additional inconsistencies.

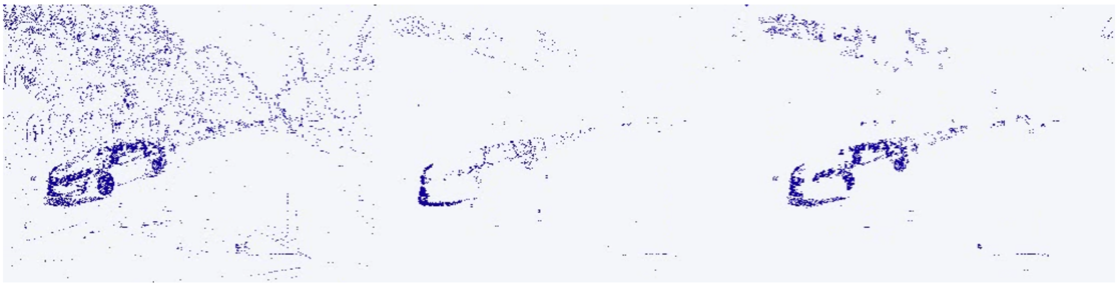


Figure 4.5: Hybrid filters are very sensitive when they are activated. Most of the time, they left a trace behind the moving object. Left is input, middle is the raw output after filter, right is post processed output

4.3.3 Clustering

The clustering stage plays a crucial role in converting sparse spike activity into coherent object level detections. This step becomes particularly important for the depressing and hybrid filters, where the interior of moving objects may be partially suppressed due to synaptic adaptation. In such cases, morphological operations provide valuable post processing by reconnecting fragmented regions and restoring the spatial completeness of the objects. These operations help compensate for the reduced activity inside the objects and allow the clustering algorithm to operate on more structured event patterns.

DBSCAN is well suited to this setting because it does not require a predefined number of clusters and can naturally handle variations in object size and spatial density. This is especially relevant in event-based scenes where the number of moving objects can change rapidly over time. However, the clustering pipeline is highly sensitive to parameter choices such as neighborhood radius, minimum samples, and post processing thresholds. As a result, extensive experimentation was required to identify stable configurations. This sensitivity creates an inherent bottleneck, since the filtering stage is designed to be adaptive, but the clustering stage requires careful manual tuning.

Another limitation arises from the fact that clustering is applied in the spatial domain only. To use DBSCAN effectively, the temporal dimension of the spikes is collapsed into the chosen event window. This simplifies the problem but discards the temporal continuity that is fundamental to neuromorphic processing. While this compromise is acceptable for practical detection tasks, it prevents the algorithm from leveraging the rich temporal structure that event cameras provide.

To address this, preliminary experiments attempted to incorporate time into the clustering process by treating events as (x, y, t) points rather than only (x, y). The idea was to preserve temporal relationships and allow DBSCAN to group events that are both spatially and temporally coherent. However, the evaluated time windows were too short to reveal meaningful temporal structure, while increasing the window duration weakened the spatial consistency of the clusters. This trade off suggests that spatio temporal clustering remains a promising direction, but further investigation is needed to determine how temporal information can be incorporated without degrading spatial accuracy.

4.4 Future Work

Several directions emerge from this work that could significantly enhance the performance and applicability of event-based motion filtering.

A first direction concerns the event data itself. In this study, all events were treated as having the same polarity, which constitutes a major simplification and leads to substantial information loss. Real event cameras produce both positive and negative polarity events, and these two channels often capture different aspects of object boundaries. Preliminary experiments indicated that the leading and trailing edges of a moving object tend to produce opposite polarities, although the exact distribution depends strongly on illumination conditions and the angle of the light source. While the incorporation of polarity did not yield notable improvements in initial trials, a deeper investigation of polarity aware filtering remains an important next step and may reveal new mechanisms for disentangling motion from background activity.

A second direction involves model architecture. The current work evaluates only single layer filtering configurations. Extending the system to multiple layers could allow the extraction of more complex motion features and enable higher level spatial abstractions. As observed, facilitating synapses directly connected to the input tend to produce uncontrolled activations, but facilitating mechanisms may become more effective when placed in deeper layers or combined with other types of synaptic dynamics. Initial experiments with stacked spatial, depressing, and facilitating layers did not yield stable improvements, but the architectural space is large and remains largely unexplored. The NEST simulator also offers a wide range of neuron and synapse models beyond the simple configuration used here, providing further opportunities for refining temporal and spatial processing.

A third direction concerns the clustering stage. While DBSCAN is practical and flexible, its reliance on spatial only clustering and its sensitivity to hyperparameters limit its adaptability. Incorporating temporal information into the clustering process showed potential, but the evaluated time windows were not long enough to reveal meaningful spatio temporal structure, while increasing the window duration weakened spatial coherence. Developing a clustering method that adheres more closely to neuromorphic principles and operates directly on spatio temporal event streams could lead to substantial improvements.

In addition to improving the clustering module itself, a further extension would be to add a dedicated classification stage at the end of the pipeline. Such a module could learn to distinguish between noise induced clusters and clusters corresponding to actual moving objects. Although this addition would increase computational cost, it could significantly strengthen the robustness of the overall framework by reducing false positives and refining the quality of detections. This represents another promising direction that has yet to be explored.

Finally, the most significant need lies in dataset development. Despite the growing interest in neuromorphic vision, current event-based datasets lack ground truth annotations that reflect the temporal and asynchronous nature of event data. Both bounding boxes and segmentation masks are generated using frame based models that do not align with the characteristics of event cameras. This mismatch creates a bottleneck for evaluating algorithms designed for motion driven or event specific processing. Since manually labeling individual events is infeasible, future datasets will require automated annotation pipelines specifically designed for event data, potentially combining segmentation models, geometric priors, and motion analysis to produce reliable ground truth.

Summary of Future Directions

- **Polarity aware processing:** Current experiments assume identical polarity for all events, resulting in information loss. Incorporating both polarities remains an important next step, especially since opposite polarities capture different motion edges.

- **Multi layer filtering architectures:** Only single layer filters were evaluated. Stacked spatial, depressing, and facilitating layers may capture richer motion features. More advanced neuron and synapse models available in NEST remain unexplored.

- **Improved spatio temporal clustering:** DBSCAN is effective but sensitive to parameters and collapses temporal information. Neuromorphic inspired clustering that operates directly on (x, y, t) events is a promising direction.

- **Post clustering classification module:** A lightweight classifier could distinguish noise induced clusters from true objects, improving robustness at the cost of additional computation.

- **Need for better event-based datasets:** Current datasets rely on frame based detectors for ground truth, which does not align with the nature of event cameras. Automated event aware annotation pipelines are essential for future progress.

Chapter 5

Conclusion

This thesis investigated a biologically inspired approach to event-based motion segmentation using SNNs and STP. The goal was to separate independently moving objects from background activity generated by ego-motion without relying on supervised learning, frame reconstruction, or heavy computational models. To achieve this, the work introduced a filtering framework that combines spatial convolutional connections with TM synapses. These synapses adapt their efficacy based on recent presynaptic activity, allowing the system to suppress repetitive background events while preserving transient, motion related ones.

The proposed framework was evaluated on two large and widely used event-based datasets: the Prophesee 1 Megapixel Automotive Dataset and the MVSEC dataset. Since these datasets do not provide event level segmentation ground truth, a dedicated post processing and clustering pipeline was designed to convert filtered spikes into bounding box predictions. The clustering relied on DBSCAN, which is well suited for event-based motion patterns because it does not require specifying the number of objects in advance.

Across all experiments, depressing synapses governed by STD dynamics emerged as the most effective mechanism for suppressing ego-motion induced background activity. Compared to static spatial filters, depressing synapses produced far fewer false positives while still preserving coherent detections of moving objects. Their adaptive behavior allowed them to downregulate sustained background activity while maintaining sensitivity to meaningful motion patterns. Hybrid filters that combined STF and STD did not provide consistent improvements, indicating that facilitation is not optimal at the first filtering layer in the current configuration. Overall, single layer filters with depressing TM synapses produced the most reliable performance across scenarios.

The interpretation of standard detection metrics required caution because ground truth bounding boxes in both datasets were generated with frame based detectors operating on APS or grayscale imagery. These annotations do not align well with the temporal resolution of event streams produced by sensors such as the DVS. As a result, precision and recall were not always reliable indicators of filtering effectiveness. The most informative metric was the number of false positives, which directly reflected how well each filter suppressed background events. From this perspective, models using depressing synapses showed the strongest robustness.

Overall, the results demonstrate that STP provides a biologically grounded and computationally efficient mechanism for event driven motion segmentation. The framework operates continuously on the event stream, does not require training, and relies entirely on local synaptic interactions. These characteristics make it suitable for future deployment on neuromorphic hardware and promising for low latency perception in autonomous systems.

Several research directions arise from this work. Polarity aware filtering could enhance motion edge discrimination. Multi layer filtering architectures may capture more complex spatial and temporal patterns. Improved spatio temporal clustering methods that operate directly on event coordinates could better exploit temporal continuity. Finally, there is a clear need for new event-based datasets with accurate ground truth that reflects the asynchronous nature of event data, since current datasets rely on frame based annotation pipelines.

In conclusion, this thesis shows that depressing TM synapses offer a principled and efficient solution for filtering event streams in motion segmentation tasks. By leveraging the adaptive dynamics of STP, the proposed framework provides a strong foundation for fully neuromorphic vision systems operating in dynamic environments.

Bibliography

[1] Dennis V Christensen et al. “2022 Roadmap on Neuromorphic Computing and Engineering”. In: *Neuromorphic Computing and Engineering* 2.2 (May 2022), p. 022501. ISSN: 2634-4386. DOI: 10.1088/2634-4386/ac4a83. (Visited on 11/10/2025) (cit. on pp. 15–17).

[2] Nitin Rathi, Indranil Chakraborty, Adarsh Kosta, Abhronil Sengupta, Aayush Ankit, Priyadarshini Panda, and Kaushik Roy. “Exploring Neuromorphic Computing Based on Spiking Neural Networks: Algorithms to Hardware”. In: *ACM Comput. Surv.* 55.12 (Mar. 2023), 243:1–243:49. ISSN: 0360-0300. DOI: 10.1145/3571155. (Visited on 11/10/2025) (cit. on p. 15).

[3] Carver Mead, Mohammed Ismail, and Jonathan Allen, eds. *Analog VLSI Implementation of Neural Systems*. Vol. 80. The Kluwer International Series in Engineering and Computer Science. Boston, MA: Springer US, 1989. ISBN: 978-1-4612-8905-0 978-1-4613-1639-8. DOI: 10.1007/978-1-4613-1639-8. (Visited on 11/10/2025) (cit. on p. 16).

[4] Amar Shrestha, Haowen Fang, Zaidao Mei, Daniel Patrick Rider, Qing Wu, and Qinru Qiu. “A Survey on Neuromorphic Computing: Models and Hardware”. In: *IEEE Circuits and Systems Magazine* 22.2 (2022), pp. 6–35. ISSN: 1558-0830. DOI: 10.1109/MCAS.2022.3166331. (Visited on 11/10/2025) (cit. on p. 16).

[5] Maximilian P. R. Löhr, Christian Jarvers, and Heiko Neumann. “Complex Neuron Dynamics on the IBM TrueNorth Neurosynaptic System”. In: *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. Aug. 2020, pp. 113–117. DOI: 10.1109/AICAS48895.2020.9073903. (Visited on 11/10/2025) (cit. on p. 17).

[6] Mike Davies, Andreas Wild, Garrick Orchard, Yulia Sandamirskaya, Gabriel A. Fonseca Guerra, Prasad Joshi, Philipp Plank, and Sumedh R. Risbud. “Advancing Neuromorphic Computing With Loihi: A Survey of Results and Outlook”. In: *Proceedings of the IEEE* 109.5 (May 2021), pp. 911–934. ISSN: 1558-2256. DOI: 10.1109/JPROC.2021.3067593. (Visited on 11/09/2025) (cit. on p. 17).

[7] Marc-Oliver Gewaltig and Markus Diesmann. “NEST (NEural Simulation Tool)”. In: *Scholarpedia* 2.4 (2007), p. 1430. DOI: 10.4249/scholarpedia.1430 (cit. on p. 17).

[8] Dhireesha Kudithipudi et al. “Neuromorphic Computing at Scale”. In: *Nature* 637.8047 (Jan. 2025), pp. 801–812. ISSN: 1476-4687. DOI: 10.1038/s41586-024-08253-8. (Visited on 11/10/2025) (cit. on p. 17).

[9] Manon Dampfhoffer, Thomas Mesquida, Alexandre Valentian, and Lorena Anghel. “Backpropagation-Based Learning Techniques for Deep Spiking Neural Networks: A Survey”. In: *IEEE Transactions on Neural Networks and Learning Systems* 35.9 (Sept. 2024), pp. 11906–11921. ISSN: 2162-2388. DOI: 10.1109/TNNLS.2023.3263008. (Visited on 11/11/2025) (cit. on pp. 18, 22, 23, 25).

[10] Zexiang Yi, Jing Lian, Qidong Liu, Hegui Zhu, Dong Liang, and Jizhao Liu. “Learning Rules in Spiking Neural Networks: A Survey”. In: *Neurocomputing* 531 (Apr. 2023), pp. 163–179. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2023.02.026. (Visited on 11/11/2025) (cit. on pp. 18, 19, 21–23, 25, 26).

[11] Paweł Pietrzak, Szymon Szczęsny, Damian Huderek, and Łukasz Przyborowski. “Overview of Spiking Neural Network Learning Approaches and Their Computational Complexities”. In: *Sensors* 23.6 (Jan. 2023), p. 3037. ISSN: 1424-8220. DOI: 10.3390/s23063037. (Visited on 11/11/2025) (cit. on pp. 18, 19, 22, 23, 25).

[12] Jason K. Eshraghian, Max Ward, Emre O. Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D. Lu. “Training Spiking Neural Networks Using Lessons From Deep Learning”. In: *Proceedings of the IEEE* 111.9 (Sept. 2023), pp. 1016–1054. ISSN: 1558-2256. DOI: 10.1109/JPROC.2023.3308088. (Visited on 11/11/2025) (cit. on p. 21).

[13] Qi Xu, Yaxin Li, Jiangrong Shen, Jian K. Liu, Huajin Tang, and Gang Pan. “Constructing Deep Spiking Neural Networks From Artificial Neural Networks With Knowledge Distillation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 7886–7895. (Visited on 11/11/2025) (cit. on p. 22).

[14] João D. Nunes, Marcelo Carvalho, Diogo Carneiro, and Jaime S. Cardoso. “Spiking Neural Networks: A Survey”. In: *IEEE Access* 10 (2022), pp. 60738–60764. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3179968. (Visited on 11/11/2025) (cit. on pp. 23, 25, 26).

[15] Kashu Yamazaki, Viet-Khoa Vo-Ho, Darshan Bulsara, and Ngan Le. “Spiking Neural Networks and Their Applications: A Review”. In: *Brain Sciences* 12.7 (July 2022), p. 863. ISSN: 2076-3425. DOI: 10.3390/brainsci12070863. (Visited on 11/11/2025) (cit. on p. 23).

[16] Jiangrong Shen, Qi Xu, Jian K. Liu, Yueming Wang, Gang Pan, and Huajin Tang. “ESL-SNNs: An Evolutionary Structure Learning Strategy for Spiking Neural Networks”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 37.1 (June 2023), pp. 86–93. ISSN: 2374-3468. DOI: 10.1609/aaai.v37i1.25079. (Visited on 11/11/2025) (cit. on p. 26).

[17] Bharatesh Chakravarthi, Aayush Atul Verma, Kostas Daniilidis, Cornelia Fermuller, and Yezhou Yang. “Recent Event Camera Innovations: A Survey”. In: *Computer Vision – ECCV 2024 Workshops*. Ed. by Alessio Del Bue, Cristian Canton, Jordi Pont-Tuset, and Tatiana Tommasi. Cham: Springer Nature Switzerland, 2025, pp. 342–376. ISBN: 978-3-031-92460-6. DOI: 10.1007/978-3-031-92460-6_21 (cit. on pp. 26, 27, 29, 31).

[18] Guillermo Gallego et al. “Event-Based Vision: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.1 (Jan. 2022), pp. 154–180. ISSN: 0162-8828, 2160-9292, 1939-3539. DOI: 10.1109/TPAMI.2020.3008413. arXiv: 1904.08405 [cs]. (Visited on 04/10/2025) (cit. on pp. 26–31).

[19] Elias Mueggler, Basil Huber, and Davide Scaramuzza. “Event-Based, 6-DOF Pose Tracking for High-Speed Maneuvers”. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Sept. 2014, pp. 2761–2768. DOI: 10.1109/IROS.2014.6942940. (Visited on 11/11/2025) (cit. on p. 27).

[20] Claudio Cimarelli, Jose Andres Millan-Romera, Holger Voos, and Jose Luis Sanchez-Lopez. *Hardware, Algorithms, and Applications of the Neuromorphic Vision Sensor: A Review*. Apr. 2025. DOI: 10.48550/arXiv.2504.08588. arXiv: 2504.08588 [cs]. (Visited on 10/31/2025) (cit. on pp. 27–31).

[21] Gianluca Amprimo, Alberto Ancilotto, Alessandro Savino, Fabio Quazzolo, Claudia Ferraris, Gabriella Olmo, Elisabetta Farella, and Stefano Di Carlo. “EHWGesture-A dataset for multimodal understanding of clinical gestures”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2025, pp. 2701–2710 (cit. on p. 28).

[22] Dario Cazzato and Flavio Bono. “An Application-Driven Survey on Event-Based Neuromorphic Computer Vision”. In: *Information* 15.8 (Aug. 2024), p. 472. ISSN: 2078-2489. DOI: 10.3390/info15080472. (Visited on 11/11/2025) (cit. on pp. 29, 30).

[23] Xu Zheng, Yexin Liu, Yunfan Lu, Tongyan Hua, Tianbo Pan, Weiming Zhang, Dacheng Tao, and Lin Wang. *Deep Learning for Event-based Vision: A Comprehensive Survey and Benchmarks*. Feb. 2023. DOI: 10.48550/arXiv.2302.08890. (Visited on 11/11/2025) (cit. on pp. 29, 30).

[24] Hao Wang, Bin Sun, Shuzhi Sam Ge, Jie Su, and Ming Liang Jin. “On Non-von Neumann Flexible Neuromorphic Vision Sensors”. In: *npj Flexible Electronics* 8.1 (May 2024), p. 28. ISSN: 2397-4621. DOI: 10.1038/s41528-024-00313-3. (Visited on 11/11/2025) (cit. on p. 30).

[25] Manish Nagaraj, Chamika Mihiranga Liyanagedera, and Kaushik Roy. *DOTIE - Detecting Objects through Temporal Isolation of Events Using a Spiking Architecture*. Oct. 2022. DOI: 10.48550/arXiv.2210.00975. arXiv: 2210.00975 [cs]. (Visited on 03/12/2025) (cit. on pp. 31, 45, 48, 53).

[26] Timo Stoffregen, Guillermo Gallego, Tom Drummond, Lindsay Kleeman, and Davide Scaramuzza. “Event-Based Motion Segmentation by Motion Compensation”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019, pp. 7243–7252. DOI: 10.1109/ICCV.2019.00734. (Visited on 11/11/2025) (cit. on p. 32).

[27] Daniel Gehrig and Davide Scaramuzza. “Low-Latency Automotive Vision with Event Cameras”. In: *Nature* 629.8014 (May 2024), pp. 1034–1040. ISSN: 1476-4687. DOI: 10.1038/s41586-024-07409-w. (Visited on 11/11/2025) (cit. on p. 32).

[28] Victoria Clerico, Shay Snyder, Arya Lohia, Md Abdullah-Al Kaiser, Gregory Schwartz, Akhilesh Jaiswal, and Maryam Parsa. “Retina-Inspired Object Motion Segmentation for Event-Cameras”. In: *2025 Neuro Inspired Computational Elements (NICE)*. Mar. 2025, pp. 1–6. DOI: 10.1109/NICE65350.2025.11065149. (Visited on 10/31/2025) (cit. on p. 33).

[29] Stefan Rotter and Markus Diesmann. “Exact Digital Simulation of Time-Invariant Linear Systems with Applications to Neuronal Modeling”. In: *Biological Cybernetics* 81.5 (Nov. 1999), pp. 381–402. ISSN: 1432-0770. DOI: 10.1007/s004220050570. (Visited on 10/29/2025) (cit. on p. 37).

[30] Markus Diesmann, Marc-Oliver Gewaltig, Stefan Rotter, and Ad Aertsen. “State Space Analysis of Synchronous Spiking in Cortical Neural Networks”. In: *Neurocomputing*. Computational Neuroscience: Trends in Research 2001 38–40 (June 2001), pp. 565–571. ISSN: 0925-2312. DOI: 10.1016/S0925-2312(01)00409-X. (Visited on 10/29/2025) (cit. on p. 37).

[31] M. Tsodyks, A. Uziel, and H. Markram. “Synchrony Generation in Recurrent Networks with Frequency-Dependent Synapses”. In: 20.1 (2000). ISSN: 1529-2401. DOI: 10.1523/JNEUROSCI.20-01-j0003.2000. (Visited on 10/29/2025) (cit. on p. 39).

[32] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD’96. Portland, Oregon: AAAI Press, Aug. 1996, pp. 226–231. (Visited on 11/05/2025) (cit. on p. 48).

[33] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. “DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN”. In: *ACM Trans. Database Syst.* 42.3 (July 2017), 19:1–19:21. ISSN: 0362-5915. DOI: 10.1145/3068335. (Visited on 11/05/2025) (cit. on p. 48).

[34] Etienne Perot, Pierre de Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. “Learning to Detect Objects with a 1 Megapixel Event Camera”. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS ’20. Red Hook, NY, USA: Curran Associates Inc., Dec. 2020, pp. 16639–16652. ISBN: 978-1-7138-2954-6. (Visited on 10/30/2025) (cit. on p. 51).

[35] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. “The Multivehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception”. In: *IEEE Robotics and Automation Letters* 3.3 (July 2018), pp. 2032–2039. ISSN: 2377-3766. DOI: 10.1109/LRA.2018.2800793. (Visited on 10/30/2025) (cit. on p. 52).

[36] Nikhila Ravi et al. *SAM 2: Segment Anything in Images and Videos*. 2024. arXiv: 2408.00714 [cs.CV]. URL: https://arxiv.org/abs/2408.00714 (cit. on p. 59).