Advancing Neuromorphic Event-Based Vision Methods for Robotic Perception Tasks

by

Zaid A. El Shair

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical, Electronics and Computer Engineering)
in the University of Michigan-Dearborn
2024

Doctoral Committee:

Associate Professor Samir Rawashdeh, Chair
Associate Professor Abdallah Chehade
Assistant Professor Jaerock Kwon
Assistant Professor Alireza Mohammadi

Zaid A. El Shair
zelshair@umich.edu
ORCID iD: 0000-0001-9518-2828

DEDICATION

I dedicate this dissertation to my wife, Mariana, for her unrelenting support and patience throughout this non-linear and challenging journey; and to my parents, Amjad and Rima, for their countless sacrifices, unwavering support, and constant encouragement that made this endeavor possible. Without them, I could not have succeeded.

ACKNOWLEDGEMENTS

First and foremost, I extend my deepest gratitude and sincere appreciation to my advisor, Dr. Samir Rawashdeh, whose guidance, patience, and numerous life lessons have been instrumental throughout my years at the University of Michigan-Dearborn. His ability to see potential in his students and unwavering support were key in my pursuit of a Ph.D. I am immensely grateful to you for encouraging me to embark on this transformative and challenging journey.

I am profoundly grateful to Dr. Paul Richardson for his guidance and support during my tenure as a Graduate Student Instructor. His chairmanship made this position possible, funding my graduate studies and enriching my academic journey. Working directly with you has been a pleasure.

Special thanks go to my colleagues and collaborators who have significantly contributed to my Ph.D. journey. Dr. Ali Hassani, for his professional mentorship and assistance during the latter part of my Ph.D., and Dr. Mohamed Aladem, for his support and guidance during the early days of my graduate program. I learned a lot from both of you. I would also like to thank my former colleague Wei Li for his friendship and support during my initial stage at UM-Dearborn. It was a pleasure working with you on my first research project as a Research Assistant.

I am thankful to my committee members, the faculty of the College of Engineering and Computer Science, and everyone I interacted with who offered advice and encouragement. This journey would not have been as enriching and successful without their collective wisdom and support.

I am deeply grateful to the ECE staff for their administrative support. A special thank you to Amanda Donovan, who has been incredibly helpful throughout my years at UM-Dearborn.

Lastly, I would like to express my heartfelt thanks to my close friends and family in Jordan, the United States, and around the world. Their support and encouragement have been a constant source of strength and motivation.

PREFACE

This dissertation represents the culmination of years of dedicated research in the field of Computer Vision, specifically focusing on Event-Based Vision for robotic perception tasks. It integrates a series of studies that have been published in various scientific journals, reflecting the progression and evolution of my research work in this domain. Each chapter of this dissertation corresponds to a specific publication, either directly adapting or building upon these published works.

Below is an outline of the connections between the dissertation chapters and their corresponding publications:

- **Chapter 2**: Detailed exploration of the MEVDT dataset developed and utilized in [45, 44].
- **Chapter 3**: Adaptation of our work titled "High-Temporal-Resolution Object Detection and Tracking Using Images and Events," published in the *Journal of Imaging*, 2022 [45].
- **Chapter 4**: Adaptation of our work titled "High-temporal-resolution event-based vehicle detection and tracking," published in *Optical Engineering*, 2023 [44].
- **Chapter 5**: Adaptation of our work titled "CSTR: A Compact Spatio-Temporal Representation for Event-Based Vision," published in *IEEE Access*, 2023 [46].
- **Chapter 6**: Builds on the developments and findings from "CSTR: A Compact Spatio-Temporal Representation for Event-Based Vision," published in *IEEE Access*, 2023 [46].

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF APPENDICES

LIST OF ACRONYMS

2D Two-dimensional

3D Three-dimensional

4D Four-dimensional

AD Automated Driving

ADAS Advanced Driver Assistance Systems

AER Address-Event Representation

AP Average Precision

APS Active Pixel Sensor

ATIS Asynchronous Time-Based Image Sensor

AVs Autonomous Vehicles

BB Bounding Box

CNN Convolutional Neural Network

CNNs Convolutional Neural Networks

CV Computer Vision

DAVIS Dynamic and Active-Pixel Vision Sensor

DL Deep Learning

DNN Deep Neural Network

DVS Dynamic Vision Sensor

FN False Negative

FP False Positive

FPS Frames Per Second

GCN Graph Convolutional Network

GCNs Graph Convolutional Networks

HDR High Dynamic Range

IoU Intersection over Union

mAP mean Average Precision

ML Machine Learning

MOT Multi-Object Tracking

NMS Non-Maximum Suppression

ROI Region of Interest

ROS Robot Operating System

SLAM Simultaneous Localization and Mapping

TN True Negative

TP True Positive

ABSTRACT

This dissertation explores the emerging field of event-based vision, a significant innovation in visual sensing technology that represents a marked departure from traditional frame-based imaging. Inspired by the biological processes of the human retina, event-based sensors operate asynchronously at the pixel level. They are characterized by their ability to capture data with high temporal resolution and exceptional dynamic range, detecting and recording changes in light intensity independently. This unique capability allows for the continuous and selective monitoring of a scene, dynamically capturing information only as necessary.

The core focus of this research is to harness the potential of neuromorphic event-based vision for advancing object detection and tracking methodologies. Despite the promising attributes of event-based sensors, their integration into conventional Computer Vision (CV) architectures poses substantial challenges, primarily due to the asynchronous and sparse nature of their output. This dissertation aims to address these challenges by developing novel methodologies that leverage the unique strengths of event-based vision while overcoming its inherent limitations.

A key contribution of this work is the introduction of the Multi-Modal Event-Based Vehicle Detection and Tracking (MEVDT) dataset. This pivotal resource, comprising synchronized streams of event data and grayscale images, facilitates the development and evaluation of novel event-based algorithms, particularly in automotive contexts. Building on this foundation, the dissertation presents a hybrid approach that integrates state-of-the-art frame-based detectors with novel event-based methods, achieving high temporal resolution in object detection and tracking. This approach is further refined with advanced techniques to enhance both detection accuracy and tracking robustness.

A central element of this research is the Compact Spatio-Temporal Representation (CSTR).

This novel representation effectively encodes event data into a format that is directly compatible with modern computer vision architectures, integrating spatial, temporal, and polarity information. The CSTR, in conjunction with a specially designed augmentation framework, significantly improves the performance of various recognition tasks.

The culmination of this dissertation is a comprehensive analysis of the CSTR and other image-like event representations in the context of event-based and multi-modal object detection. Rigorous testing on two event-based multi-modal datasets demonstrates the effectiveness of these methods, offering insights into their comparative performances and the synergies between event-based and frame-based sensors. Through these comprehensive evaluations, this work underscores the importance of optimal spatio-temporal representations for event-based vision tasks. Ultimately, this dissertation represents a step towards the practical application of event-based vision, contributing to the ongoing evolution in the field of CV.

CHAPTER 1

Introduction

1.1 Research Motivation and Overview

1.1.1 Perception in Robotics and Automation

CV is a critical component in robotics and automation, transforming digital data into actionable insights about the environment. This transformative ability is essential for autonomous decision-making, where an accurate and timely understanding of the surroundings dictates the efficacy of the robotic actions [34, 47]. Recent advancements in CV, particularly through Deep Learning (DL), have propelled the field towards achieving, and in some cases surpassing, human-level performance in crucial perception tasks [72, 92, 149, 159]. However, the limitations inherent in traditional visual input systems, such as limited dynamic range and sensitivity to environmental conditions, still pose challenges in complex and dynamic settings. These constraints underscore the necessity for exploring more advanced perception technologies that can handle diverse and unpredictable environments with greater reliability. The development of such technologies promises to bridge the gap between current capabilities and the requirements of fully autonomous systems.

1.1.2 Common Sensors in Perception

In domains like Automated Driving (AD) and Advanced Driver Assistance Systems (ADAS), a diverse array of sensors are employed to inform machine vision. These sensors typically include

LiDARs, radars, and digital cameras, often based on CMOS technology [6]. LiDARs, valued for their precision in distance measurements and high-spatial-resolution range information, can be limited by high costs and susceptibility to certain weather conditions [177]. Radars offer robustness in adverse weather conditions such as fog but are constrained by their limited resolution [61]. Traditional digital cameras, typically CMOS-based, provide detailed visual information but face challenges with dynamic range and low-light conditions [109, 116, 139, 73].

The integration of these sensors is crucial for reliable decision-making, as each sensor type compensates for the shortcomings of others. For instance, fusion techniques, especially between LiDAR and traditional cameras, have shown promise in enhancing object detection and classification in real-time applications for Autonomous Vehicles (AVs) [10]. However, the integration of these diverse technologies into a unified perception system introduces challenges, including data fusion and calibration complexities [184]. This integration is often further constrained by factors such as cost and reliability in various applications.

Traditional cameras, often CMOS-based, are frequently favored in many robotic and automation systems due to their resemblance to human visual perception, as well as their broad availability and low cost. However, this reliance also brings to light the inherent limitations of these cameras, underscoring the need for a multi-sensor approach to achieve comprehensive perception in AD and ADAS [127, 168].

1.1.3 Advancements in Computer Vision Driven by Deep Learning

In the past decade, CV has undergone a paradigm shift, largely driven by breakthroughs in DL [88, 92]. This transformative shift in Machine Learning (ML) has enabled models to bypass the need for manual feature extraction [48], allowing for direct learning from raw sensor data [120]. Particularly, deep Convolutional Neural Networks (CNNs) have revolutionized tasks such as image classification, significantly outperforming previous benchmarks [186]. This revolution in CV is partly due to increased computational power and the availability of extensive, labeled datasets, establishing DL as the predominant method in contemporary CV applications [1, 67, 71].

However, these advancements in DL are intrinsically linked to the quality of the input data. The performance of DL models in CV is often constrained by the limitations of the sensors providing this data. For instance, traditional cameras, with their limited dynamic range and sensitivity to varying lighting conditions, can restrict the effectiveness of DL in complex visual tasks [30, 69, 118]. This highlights the ongoing need for improvements not only in DL algorithms but also in sensor technology, to fully harness the potential of advanced CV systems in automated applications.

1.1.4 Limitations of Contemporary Vision Sensors in Automation

Traditional vision sensors, predominantly CMOS cameras, are integral to AD and ADAS systems due to their cost-effectiveness and ability to provide dense visual information akin to human perception. Despite their widespread use, these cameras are limited by factors such as limited dynamic range, high computational demands, susceptibility to motion blur, and limited update rates[28]. These limitations can lead to suboptimal performance in critical scenarios, as evidenced by several real-world incidents in AD systems [125, 164].

In 2016, a Tesla Model S with Autopilot engaged (Tesla's version of ADAS) collided with a white 18-wheeler truck, resulting in a fatal crash [164] (incident illustrated in Figure 1.1). The Autopilot system failed to detect the truck against a bright sky, a situation exacerbated by the camera's low dynamic range as visualized in Figure 1.2. This incident underscores the challenges cameras face in distinguishing objects in overexposed scenes.

In 2018, an Uber AD test vehicle struck a pedestrian crossing a street at night [125] (incident illustrated in Figure 1.3). Despite the pedestrian reportedly being detected by the vehicle's LiDAR [128], the system misclassified it as a false positive, influenced by the limited dynamic range of the camera used in low-light conditions. This accident highlights the challenges cameras face in detecting objects in poorly lit low-contrast environments as demonstrated in Figure 1.4.

These incidents illustrate the inherent limitations of relying heavily on traditional cameras in ADAS and AD applications. While the dense visual output of cameras is crucial for detailed scene

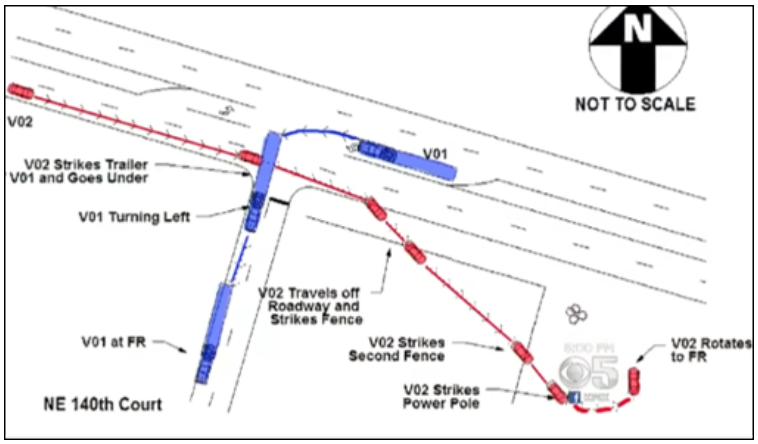


Figure 1.1: Illustration of the Tesla Autopilot accident's events[1].

Figure 1.2: Visualization of the camera's output right before the accident[2].

information, their limitations in diverse and dynamic environments highlight the need for a more balanced and robust perception approach. Current multi-modal sensor systems, integrating cameras with LiDARs and radars, attempt to address these challenges but still encounter limitations, especially in complex and unpredictable scenarios.

This situation underscores the pressing need for exploring alternative sensing technologies that can offer more robust and adaptive visual perception. Such technologies must not only handle typical scenarios effectively but also excel in edge cases, ensuring safety and reliability in autonomous

[1]Image source: YouTube https://www.youtube.com/watch?v=s8AHzY7xr10

[2]Photoshopped version of the image: *transportation truck on the road Stock Photo by mblach* – PhotoDune https://photodune.net/item/transportation-truck-on-the-road/25819561

Figure 1.3: Illustration of the accident's events and the aftermath of vehicle used[3].

Figure 1.4: Uber automated test vehicle's camera output moments before the crash[4].

applications. This need forms the basis for investigating event-based vision as a promising alternative, potentially bridging the gap between current capabilities and the high demands of real-time, efficient perception systems in dynamic environments.

1.1.5 Motivation and Problem Statement

The constraints of conventional camera technologies, along with the limitations of other sensing modalities, highlight the necessity for a more advanced and adaptive visual perception system. These systems must be capable of handling edge cases and ensuring safety and reliability in autonomous applications. Despite the strides made in CV, the shortcomings of current sensor tech-

[3]Image source: https://www.spri.kr/webroot/lib/fileman/Uploads/post_images/2018_07/2018_07_03_01.png
[4]Video source: YouTube https://www.youtube.com/watch?v=q7d90ZFhg28

nologies hinder their ability to match the efficiency and real-time capabilities of biological visual systems. This gap in performance and the quest for robust, low-latency perception systems in dynamic environments motivate the exploration of innovative sensing technologies, such as event-based vision, which promise to overcome these challenges.

1.2 Introduction to Event-Based Vision

1.2.1 Event-Based Vision: An Overview

Event-based vision is an emerging field in visual sensing technology, characterized by a novel approach to capturing and processing visual information. Drawing inspiration from the biological processes of the human eye [141], particularly the retina, event-based vision systems represent a paradigm shift from traditional frame-based imaging techniques.

At the core of event-based vision are event-based sensors, also known as *neuromorphic sensors* and *event cameras*. Unlike conventional cameras that capture a sequence of frames at regular intervals, event cameras operate asynchronously at the pixel level. They are designed to detect and record changes in intensity for individual pixels independently. Each pixel in an event camera functions autonomously, generating data—referred to as ”events”—only when a change in light intensity is detected. Each event provides information about the time, location, and polarity of the brightness change, offering a continuous asynchronous stream of information about the visual scene.

The fundamental principle behind event-based vision is its focus on capturing the dynamic aspects of a scene. Traditional cameras record the entire scene at fixed time intervals, regardless of whether changes have occurred or not. In contrast, event cameras are attuned to changes, making them highly responsive to motion and temporal variations in a scene.

Event-based vision systems are gaining attention for their potential to provide more efficient and effective visual processing, especially in environments where speed and responsiveness are crucial. This approach to vision technology offers unique capabilities, which are being explored for a range

of applications from robotics and autonomous vehicles to augmented reality and surveillance.

As research in this domain advances, event-based vision is poised to offer new perspectives in CV, challenging conventional methods and providing innovative solutions to complex visual processing tasks.

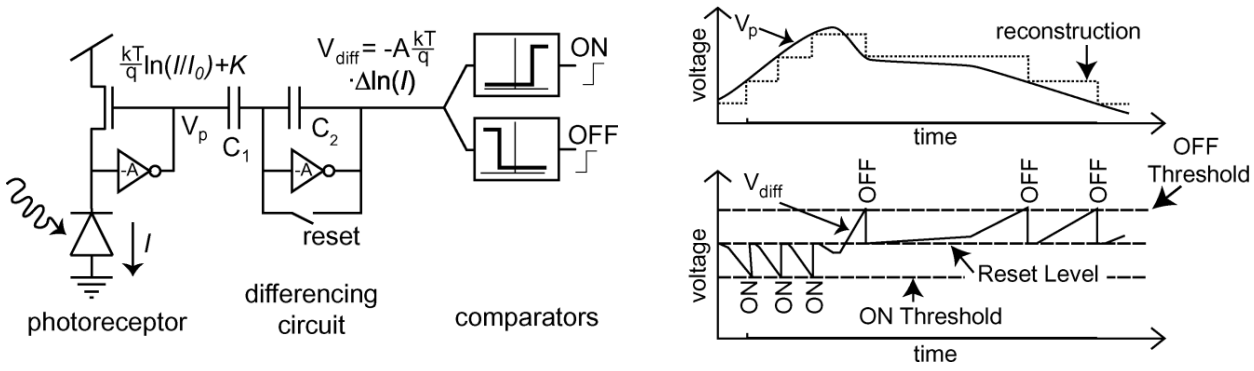
1.2.2 Historical Evolution of Neuromorphic Event-Based Vision

The concept of event-based vision originated from the desire to mimic the human retina's way of processing visual information [112, 141]. Early research in the field mainly aimed to demonstrate biological models without a focus on their practical real-world applications [112, 141]. Later on, the research focus in this field shifted towards practical perception applications motivated by the limitations of traditional frame-based cameras, particularly their inability to efficiently handle dynamic and fast-changing environments. Pioneering work in neuromorphic engineering, a field that aims to replicate neural systems' structure and functionality, laid the groundwork for the development of event-based sensors [107].

The first practical and commercially available event-based sensor was the Dynamic Vision Sensor (DVS)-128 [101]. This sensor marked a significant milestone in the field of event-based vision. The DVS-128, first made available in 2008, provides a resolution of 128×128 pixels and a dynamic range of 120 dB. This has influenced the development of other sensors that focused on addressing some of its shortcomings, such as the Asynchronous Time-Based Image Sensor (ATIS) [140], in 2010; and the Dynamic and Active-Pixel Vision Sensor (DAVIS) [21], in 2014.

Over the years, event-based vision technology has seen numerous advancements. Key milestones include the improvement in sensor resolution, the increase in the dynamic range, and the reduction in latency and power consumption [100, 101, 140, 21, 162]. These developments have expanded the potential applications of event-based vision, moving it from a theoretical concept to a practical tool in various fields.

The historical evolution of event-based vision not only highlights the technological progress in the field but also underscores the growing recognition of its potential to revolutionize how ma-



(a) abstracted pixel core schematic

(b) Principle of operation

Figure 1.5: Abstracted circuitry of a single DVS-128 pixel and illustration of its principle of operation. Reprinted from [101] © 2008 IEEE.

chines perceive and interact with their environment.

1.2.3 Technical Foundations of Event-Based Sensors

Event-based neuromorphic sensing is grounded in a fundamentally different approach to visual perception, one that captures the essence of dynamic changes in the environment. The key principles of event-based sensing are centered around the unique functionality of event cameras and their method of data acquisition and processing.

In contrast to traditional cameras that capture entire frames at fixed time intervals, each pixel in an event camera operates independently and asynchronously. This means that pixels in an event camera are not synchronized to a global shutter. Instead, they react individually to changes in light intensity, generating data whenever a change exceeds a predefined threshold. This asynchronous approach allows for the continuous monitoring of a scene, with the camera capturing information only when and where it is needed.

1.2.3.1 Functional Components of a DVS Pixel

A single DVS pixel [101] consists of 3 main components: the photoreceptor, the differencing circuit, and the comparators. These components, along with the general DVS design, are demonstrated in Figure 1.5(a).

The photoreceptor is responsible for converting the incident light into an electrical signal V_p. It features a logarithmic response, automatically controlling the pixel's gain while responding quickly to changes in illumination. However, this design leads to a DC mismatch between pixels due to transistor threshold variation, which requires calibration for direct output use.

The differencing circuit is connected to the photoreceptor and amplifies changes in the signal with high precision. It plays a crucial role in removing the DC mismatch by balancing the output to a reset level after an event is generated, as shown in Figure 1.5(b). The gain of this change amplification is determined by the well-matched capacitor ratio C_1/C_2. This precise gain helps reduce the impact of comparator mismatch.

Finally, the comparators are simple two-transistor components that compare the output V_{diff} of the differencing circuit against a reference level to generate events. They are essential for converting the analog signal into a digital output (as either ON or OFF events), indicating changes in the scene's illumination.

Together, these components allow the DVS pixel to be sensitive to temporal contrast, defined in [101] as:

$$TCON = \frac{1}{I(t)}\frac{dI(t)}{d(t)} = \frac{d(ln(I(t)))}{dt}, \tag{1.1}$$

where I is the photocurrent. This sensitivity to temporal contrast, rather than absolute light levels, enables the DVS to efficiently detect and respond to changes in a scene, making it highly suitable for dynamic and high-speed applications.

1.2.3.2 Combining DVS with Active Pixel Sensor (APS)

While the DVS excels in dynamic and high-speed scenes, it is limited in applications of a static nature. Since DVS pixels capture only relative changes in brightness, they lack the capability to provide information on the static light intensity, a limitation that the ATIS was developed to address [140]. The ATIS added an APS circuitry to capture static light intensity as well. However, the ATIS's dependency on the DVS to trigger light-intensity updates results in asynchronous and non-uniform exposure times, leading to potential motion artifacts [21].

Another groundbreaking development is the DAVIS [21]. The DAVIS was also introduced to address the DVS's limitation by combining the DVS with an APS at the pixel level, sharing the same photodiode [21] (as demonstrated in Figure 1.6). Here, the DVS part operates as described earlier while the APS component functions independently by continuously measuring and integrating the photocurrent over time to produce a voltage signal representing the static light intensity [21, 141]. This is done using differential double sampling (for the APS readout), essential for removing significant fixed pattern noise which is a common issue in CMOS-based image sensors [23].

Differential double sampling is a critical function of the APS readout that is done by taking two measurements of V_{aps}. These measurements are referred to as the *reset* voltage and the *signal* voltage. The reset voltage is sampled immediately after the pixel is reset, serving as a reference point. The signal voltage is taken at the end of the exposure period, which represents the accumulated light charge in the pixel due to light exposure. The difference between these two voltage samples is proportional to the amount of light that has struck the pixel during the exposure time, which is used to determine the light intensity. Finally, the resulting analog value is converted to a digital signal which is then used to form an image.

This setup allows the DAVIS to capture conventional images with intensity encoding, compatible with established CV research, in addition to the asynchronous event data. The APS shares the same photodiode with the DVS circuit, only adding a small readout circuit and minimally increasing the pixel area [21].

In its initial design, the DAVIS utilized a rolling shutter which led to motion artifacts in dynamic scenes, similar to those observed with the ATIS [21]. To overcome this challenge, the DAVIS was later improved by incorporating a global shutter, a feature commonly found in CMOS-based imaging sensors. This advancement substantially minimizes motion artifacts compared to both the initial DAVIS and the ATIS designs. Unlike the ATIS, which relies on asynchronous event-triggering for pixel intensity updates, the DAVIS provides a simultaneous output of asynchronous events and synchronous frames in both rolling and global shutter modes. This design allows for

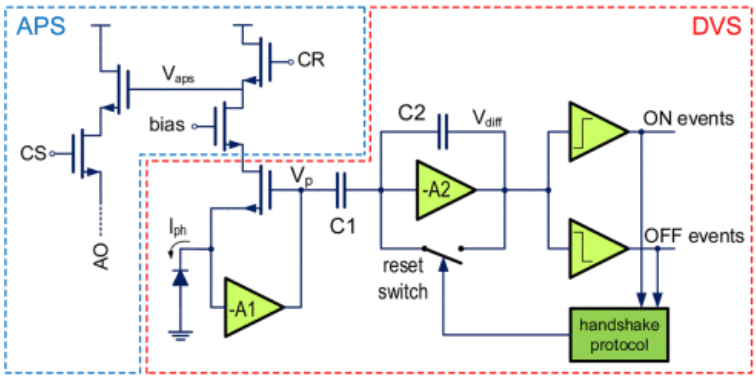


Figure 1.6: Simplified schematic of the DAVIS pixel which combines the DVS and the APS into one. Reprinted from [141] © 2014 IEEE.

more consistent and uniform exposure times across the sensor array, enhancing the sensor's ability to accurately capture both dynamic and static elements of a scene.

Overall, the integration of the DVS and APS in the DAVIS sensor broadens the range of applications for this imaging technology. The DVS addresses dynamic and high-speed components of a scene, whereas the APS enables high-accuracy object recognition and the application of established CV methods.

1.2.3.3 Address Event Representation

Given that an event-based sensor would include thousands of pixels [21, 101], a feasible wiring mechanism with a compatible communication protocol is required [141]. The Address-Event Representation (AER) is a communication protocol that simplifies the design quite significantly [107]. By combining an array of sensing nodes (such as the DVS pixel) with a common bus, the AER allows each pixel in the array to operate independently and asynchronously [101, 107]. Each event is encoded as an address (*i.e.*, an (x, y) coordinate) that uniquely identifies the pixel's location in the sensor array. The event-driven nature of AER means that data is only transmitted when changes are detected, leading to efficient data transmission and reduced power consumption compared to continuously streaming the entire pixel array's state. An event that occurs at a given pixel would be reported as location, time of occurrence, and polarity of the brightness change.

This asynchronous communication method is crucial for capturing the temporal dynamics of a

scene efficiently. AER's event-driven approach ensures that only relevant data—those parts of the scene undergoing change—are transmitted, significantly reducing the volume of data compared to traditional frame-based systems. This efficiency is particularly beneficial in dynamic environments with sparse activity, where AER can drastically cut down on unnecessary data transmission.

Furthermore, AER's design addresses scalability challenges [141]. It enables the integration of large numbers of pixels in a sensor array without overwhelming the system's bandwidth. This scalability is essential for developing high-resolution event-based sensors capable of capturing detailed visual information.

AER also facilitates the integration of event-based sensors with external processing units, such as FPGAs or digital processors [141]. This integration is key for building comprehensive vision systems that combine the unique capabilities of event-based sensing with advanced computational methods.

In summary, AER plays a pivotal role in the operation of event-based vision systems. Its efficient, scalable, and asynchronous data transmission protocol makes it an indispensable component in applications requiring real-time processing and responsiveness, such as in robotics and autonomous vehicles

1.2.4 Key characteristics and Advantages

Event-based sensors have many unique properties that differentiate them from other modalities, including:

- ***High Temporal Resolution***: Each event is captured and timestamped with microsecond precision (1 μs), leading to potential output rates up to 1 MHz. This high temporal resolution, enabled by a digital readout process where event timings are self-encoded [141], effectively eliminates motion blur issues common in traditional cameras due to prolonged exposure times.

- ***HDR***: Event cameras offer an impressive dynamic range exceeding 120 dB. Their loga-

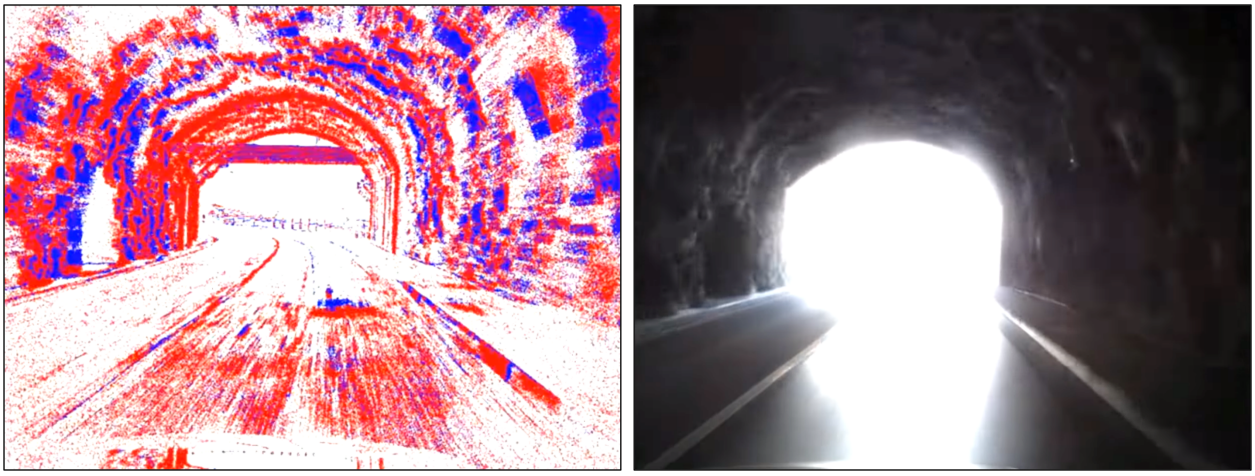


Figure 1.7: Comparison between the High Dynamic Range (HDR) output of event cameras (left) and the limited dynamic range output of frame-based cameras (right) in an HDR scene of a vehicle exiting a tunnel. The presented sample is obtained from the event-based dataset, DSEC [59], captured using Prophesee Gen 3.1[5].

rithmic response to varying light intensities allows them to compress a wide range of light levels efficiently. Coupled with the independent operation of each pixel, this feature enables the sensor to capture detailed imagery in scenes with both very bright and very dark areas [21, 101]. In contrast, traditional cameras have a limited dynamic range of ∼60 dB, struggling in such varied lighting conditions. The superior dynamic range of event cameras is visually demonstrated in Figure 1.7, highlighting their ability to capture details missed by conventional CMOS-based cameras.

- ***Low Latency***: Event-based sensors demonstrate minimal output latency, typically in the range of a few microseconds under most lighting conditions [21]. This low latency results from fast photoreceptor circuits, efficient processing, independent pixel operation, and optimized data transmission protocols. This characteristic is particularly beneficial for time-sensitive applications such as AVs.

- ***Low Power Consumption***: Depending on the DVS's activity, the power consumption of event cameras is typically in the milli-watt range (*e.g.*, 5–14 mW for the DAVIS [21]). This

[5]As visualized in the video available at: https://www.youtube.com/watch?v=uTa6-ME547Q&t=17s

efficiency stems from their event-driven nature, where power is consumed only when pixels detect brightness changes, in contrast to the consistently high power requirements of traditional frame-based cameras due to their synchronous and often redundant output.

These properties enable event-based cameras to offer robust performance in various challenging scenarios, including those with HDR, rapid motion, and low-light or night-time conditions. Their ability to provide detailed, spatially-dependent output, akin to traditional frame-based cameras, sets them apart from sensors like LiDAR and radar, which typically produce sparser outputs. This characteristic makes event cameras not only a valuable addition to existing visual sensing arrays but also viable candidates for primary visual input sensors in various applications.

1.2.5 Applications of Event-Based Vision

Event-based vision has found a diverse array of applications across various domains, leveraging its unique capabilities to address complex challenges. Some notable uses include:

- *Low-Power Monitoring and Surveillance*: Event cameras are increasingly used in monitoring systems due to their low power consumption, making them ideal for prolonged surveillance tasks [19].

- *High-Speed Obstacle Detection and Avoidance*: In dynamic environments, event-based sensors excel in detecting and avoiding obstacles at high speeds, crucial for applications like drone navigation and robotics [51, 52].

- *Deblurring Videos*: The high temporal resolution of event cameras allows them to effectively deblur videos, a capability beneficial in enhancing video quality and clarity [132].

- *Advanced SLAM*: Incorporating event cameras into Simultaneous Localization and Mapping (SLAM) systems enables more robust and accurate navigation, particularly in high-speed and high dynamic range scenarios, by utilizing a combination of images, inertial data, and event data [51, 52].

These applications demonstrate the transformative potential of event-based vision in various fields. The novel nature of event cameras has opened up new avenues for research and development, expanding the frontiers of what is possible in CV and autonomous systems.

1.2.6 Challenges and Research Opportunities

While event-based vision presents a paradigm shift in visual sensing, it is still evolving, facing both technical challenges and untapped research potential. Current challenges in the field include:

- *Utilization of High-Temporal-Resolution Data*: A key challenge is effectively harnessing the high-temporal-resolution data provided by event-based sensors. This involves developing algorithms and systems that can interpret and process data at microsecond scales for real-time applications.

- *Integration with Traditional Vision Systems*: Bridging the gap between asynchronous event-based data and traditional frame-based vision systems remains a significant hurdle. This includes finding effective ways to represent and process event data to leverage advancements in conventional CV techniques.

- *Lack of Standardized Datasets and Labeling Techniques*: The absence of large, labeled datasets for event-based vision hampers the development of DL-based solutions. Establishing standardized datasets and labeling methods is crucial for advancing machine learning approaches in this domain and for benchmarking the performance of event-based systems, especially in high-speed scenarios.

Besides these challenges, the field of event-based vision is filled with research opportunities:

- *Development of Application-Specific Solutions*: There is vast potential for creating tailored solutions that leverage the unique properties of event-based sensors (*e.g.*, HDR and low latency) in specific applications, such as AVs, robotics, and augmented reality.

- *Advancement in Sensor Technology and Cost Reduction*: As the technology matures, there is scope for reducing sensor costs and making them more accessible for mainstream applications, which in turn would generate more data for research and development.

- *Exploration of multi-modal Sensory Integration*: Integrating event-based vision with other sensory modalities presents an exciting avenue for creating more robust and efficient perception systems. This integration could lead to breakthroughs in how machines perceive and interact with their environment.

Addressing these challenges and exploring these research avenues will significantly advance the field of event-based vision, paving the way for innovative applications and enhancing the capabilities of autonomous systems.

1.3 Research Objectives and Dissertation Scope

1.3.1 Overall Research Goals and Objectives

This dissertation aims to push the boundaries of event-based vision for object detection and tracking. The objectives are to:

1. **Develop Robust Event-Based Methodologies:** Create innovative approaches for object detection and tracking using event-based data, addressing challenges and leveraging its unique advantages.

2. **Integrate Event-Based and Frame-Based Vision** Investigate hybrid methodologies integrating event-based and frame-based vision, utilizing their combined strengths for improved detection and tracking.

3. **Advance Event-Based Vision Research** Use specialized datasets to promote research in event-based vision, particularly focusing on object detection and tracking in automotive and robotic applications.

1.3.2 Chapter-Specific Aims

The dissertation unfolds through several chapters, each with distinct objectives and contributions to the field of event-based vision. In Chapter 2, the focus is on the introduction and detailing of the MEVDT dataset. This dataset serves as a pivotal tool for advancing research in multi-modal event-based object detection and tracking. Chapter 3 explores a hybrid approach for high-temporal-resolution object detection and tracking. This chapter leverages state-of-the-art frame-based detectors and introduces novel event-based methods to enhance the overall methodology. Building upon this, Chapter 4 further refines these methodologies by incorporating advanced event-based techniques, aiming to improve both the accuracy and robustness of detection and tracking. In Chapter 5, the dissertation presents the development of the CSTR, a new compact spatio-temporal representation tailored for event-based vision. This chapter assesses the CSTR's effectiveness in integrating sparse event data with conventional CV networks, particularly in the context of object and action recognition tasks. Finally, Chapter 6 delves into the application of the CSTR and other event representations in event-based and multi-modal object detection, rigorously testing these methods on two distinct event-based multi-modal datasets.

1.4 Dissertation Outline

The dissertation comprises the following chapters:

- **Chapter 1**: Sets the foundation for the dissertation by introducing the field of event-based vision and its significance.
- **Chapter 2**: Introduces the MEVDT dataset, emphasizing its importance in event-based object detection and tracking.
- **Chapter 3**: Presents a hybrid approach for high-temporal-resolution object detection and tracking, leveraging both frame-based detectors and event-based methods.

- **Chapter 4**: Expands upon Chapter 3 by incorporating advanced event-based techniques to enhance detection and tracking accuracy and robustness.

- **Chapter 5**: Discusses the development of the CSTR, a novel representation for integrating event data with conventional CV techniques.

- **Chapter 6**: Explores the application of CSTR and other image-like representations in event-based and multi-modal object detection, investigating the integration of frame-based and event-based vision.

- **Chapter 7**: Provides a consolidated conclusion and general discussion, synthesizing the findings of this dissertation. It outlines the overall contributions and discusses future research directions.

CHAPTER 2

Multi-Modal Event-Based Object Detection and Tracking Dataset

In this chapter, we introduce the **MEVDT** dataset: **M**ulti-modal **E**vent-based **V**ehicle **D**etection and **T**racking dataset. This dataset provides a synchronized stream of event data and grayscale images, captured using the novel DAVIS 240c hybrid event-based camera. It features manually annotated ground truth data, including object class, bounding boxes, and unique object IDs, at a labeling frequency of 24 Hz. Designed to advance research in event-based vision, MEVDT addresses the critical need for high-quality, annotated datasets that enable the development and benchmarking of object detection and tracking algorithms in automotive environments.

2.1 Introduction

Event-based vision represents a paradigm shift in visual sensing technology, where sensors, inspired by the biological processes of the human retina, capture dynamic changes in a scene at high temporal resolution [101, 140, 141]. Unlike traditional frame-based cameras, event-based sensors asynchronously report per-pixel brightness changes, offering advantages in dynamic range and temporal resolution [55]. This emerging field necessitates specialized datasets to promote research and development, particularly in CV tasks of object detection and tracking.

Object tracking stands as a critical task in various robotic applications, including AD and traffic monitoring [45, 76, 115]. Event-based vision, due to its low latency and high-temporal-resolution

output, offers promising prospects in these areas. However, the development of methodologies for these applications has been hampered by a lack of labeled event-based datasets. While there are numerous datasets for traditional vision applications [60, 88], a notable gap exists in labeled event-based datasets specifically tailored for object tracking. Existing event-based datasets, while useful for traditional CV tasks, are particularly lacking in the object tracking application that requires annotations that include object IDs. This gap limits the exploration and development of event-based object-tracking methods.

To address this limitation, we have created a new dataset specifically tailored for object detection and tracking in event-based vision. While our dataset does not encompass highly dynamic scenarios, it includes sequences with multiple simultaneous vehicle objects moving at various speeds. A key feature of our dataset is the provision of object IDs along with bounding box annotations, a deficiency in existing datasets. This inclusion is crucial for enabling object tracking evaluation, which forms a significant part of our research.

Our dataset[1], though not characterized by diverse scenarios, offers a compact and straightforward environment with accurate and precise labeling. This simplicity makes it an ideal starting point for researchers to develop, evaluate, and refine various event-based and hybrid methodologies. The dataset's utility is demonstrated across different chapters of this dissertation, highlighting its versatility for both developing novel solutions and for the application of the training and fine-tuning of DL-based solutions.

By offering a dataset with detailed annotations and a specific focus on vehicle-type objects, we aim to stimulate research in event-based vision, particularly in applications demanding high-speed perception and accurate object tracking. This dataset is intended as a foundational resource for researchers exploring the innovative and rapidly developing field of event-based vision.

[1]Dataset is available at http://sar-lab.net/event-based-vehicle-detection-and-tracking-dataset/

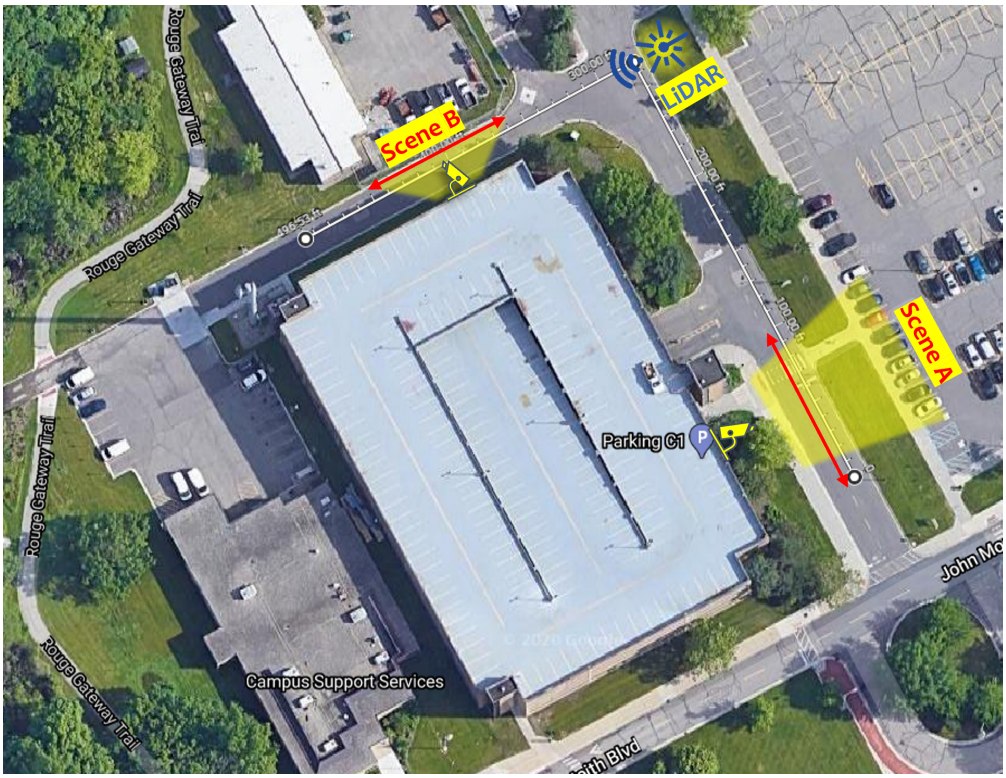


Figure 2.1: Satellite view of a subsection of the University of Michigan-Dearborn campus highlighting Scene A and Scene B, where data was collected, along with the position of the LiDAR sensor.

2.2 Dataset Collection and Labeling Method

2.2.1 Data Collection Setup

This section outlines the data collection setup for our event-based vision dataset, detailing the sensors, settings, and locations employed.

We use the hybrid sensor DAVIS 240c[2] [21], which combines an APS as well as DVS using the same pixel array. This sensor is selected as it is able to capture both asynchronous events and synchronous frames which are needed for developing event-based and multi-modal solutions and to enable accurate data labeling. The spatial resolution of this sensor is 240×180 pixels. The APS of this sensor captures intensity (*i.e.*, monochrome) frames at a fixed rate of ~ 24 Frames Per Second (FPS). Meanwhile, the DVS, can capture events, asynchronously, and at high temporal resolutions

[2]DAVIS 240c specifications available at https://inivation.com/wp-content/uploads/2019/08/DAVIS240.pdf

Scene A **Scene B**

Figure 2.2: Sample image outputs from the dataset demonstrating the two distinct scenes captured. Scene A is depicted on the left, and Scene B on the right, showcasing the camera's perspective and field of view for each location within the University of Michigan-Dearborn's campus.

of 1 μs. The fundamental concepts of DAVIS are detailed in Section 1.2.3. Additionally, an industrial high-speed LiDAR Benewake TF03-100 was employed in a subset of the data collection process to provide high-temporal-resolution ground truth positional measurements. Specifically, this LiDAR is used to estimate the distance to the car being tracked with timestamps at very high rates (up to 1000 Hz). The LiDAR is placed at a range of 60-30 meters away from the vehicle driving towards it. This is covered in more detail in Chapter 4.

Using DAVIS 240c and the Robot Operating System (ROS) DVS package developed by Robotics and Perception Group[3] [122] to record the data, we collect several hours of spatiotemporally synchronized images and events. The data collection is conducted at two different places within the same location (at the campus of the University of Michigan-Dearborn), referred to as *Scene A* and *Scene B*. Each scene was recorded on a different day with generally clear daylight conditions. A satellite map view depicting the data collection location including the positions of each scene is shown in Figure 2.1. Furthermore, we demonstrate each scene using some of the captured grayscale images in Figure 2.2.

During the data collection process, the event camera was placed on the edge of a building while pointing downward at the street, representing an infrastructure or traffic surveillance camera

[3] Available at https://github.com/uzh-rpg/rpg_dvs_ros

Figure 2.3: The data collection setup showing the hybrid event camera (DAVIS 240) mounted on a tripod at the edge of a building overlooking the street and part of the parking lot. A laptop adjacent to the camera setup is used for data recording and sensor control.

setting, as demonstrated in Figure 2.3. The camera is fixed and kept static throughout (*i.e.*, no ego-motion is applied to the camera). Accordingly, the events captured would be only due to an object's motion or due to noise. Additionally, the standard lens, shipped with the sensor, is tuned to enable viewing angles and fields of view as shown in Figure 2.1. In this dataset, we focus on capturing sequences of moving vehicles of different types (*e.g.*, sedans, trucks, etc.), as shown in Figure 2.4. Some data on pedestrians passing by is also collected (in Scene A) but is not the focus of this work due to their relatively slow movements and their far proximity to the camera, making the CV task of object detection challenging and intermittent. We also note that the vehicles that passed by in the scene did so at varying speeds and accelerations, some reaching a full stop at several instances, thus making the tasks of object detection and tracking more challenging when only using the event data (event-based vision modality). Finally, the vehicles parked at the top in Scene A, are not labeled due to them being static and due to their relative size, including any vehicles moving behind them. Overall, the top 15–20% of the frame's height can be ignored for the desired perception task.

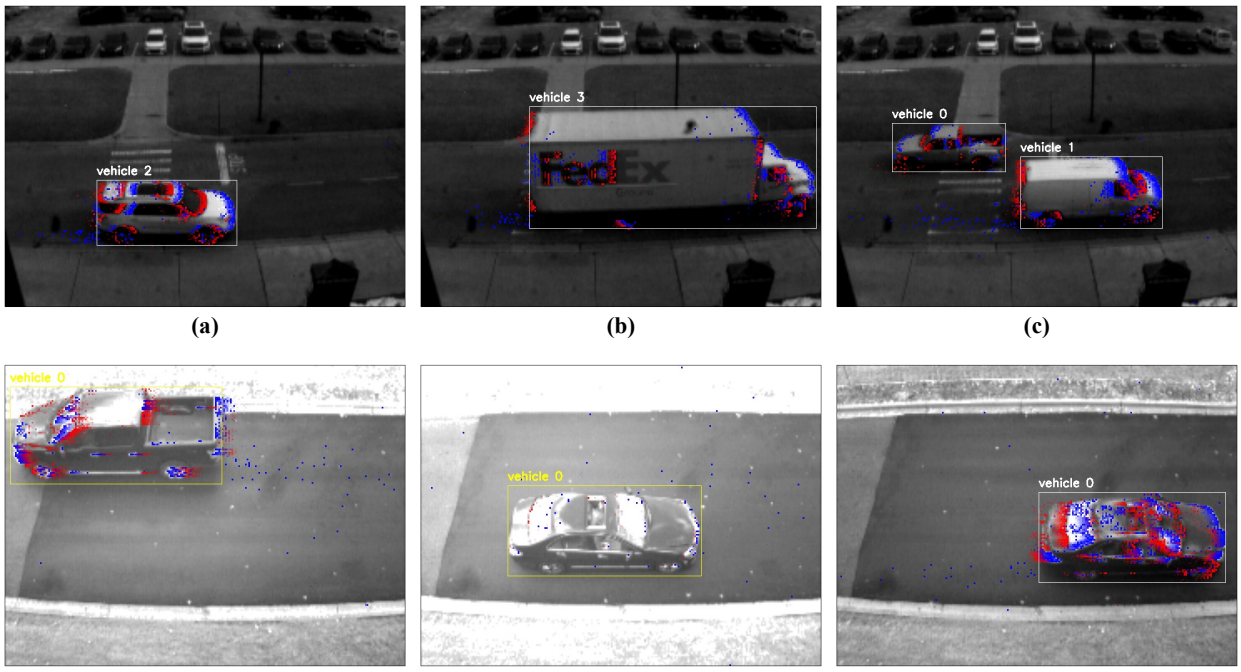


Figure 2.4: Samples from the dataset showing labeled vehicles. Each image combines the APS intensity frame with superimposed events from the DVS collected in the last ∼43 ms. The samples include various vehicle types such as SUVs (a), trucks (b), vans (c), and pickup trucks (d) captured in two different scenes (Scene A in the top row, Scene B in the bottom row). The presence of multiple objects and vehicles at different speeds (e-f) illustrates the dataset's utility for object detection and tracking research.

2.2.2 Data Processing and Labeling

Initially, we split the recorded data into short sequences. This is done in order to minimize the intervals and samples without objects present in the scene. Accordingly, the recordings of each scene are split into ∼30 short sequences, maintaining intervals with objects present in the scene while removing intervals that did not contain any.

Labeling was performed on intensity images generated by the APS, using the *dLabel Annotation Tool*[4]. After extracting the frames from each sequence, we carefully label each image by manually annotating all the *vehicle*-type objects available in each image using Two-dimensional (2D) bounding boxes and providing a unique ID. We ensure that each bounding box is labeled at sub-pixel accuracy. This results in a labeling frequency of $\sim$ 24 Hz matching the framerate of the

[4]Available at https://dlabel.org/

sensor's APS. Labels are directly transferable to the event-based modality, thanks to temporally synchronized event data matched with image timestamps. Thus, a label at a given timestamp can be used for both modalities. This is due to both sensor types using the same lens, meaning that they are spatially synchronized. A change that occurs at pixel (x_i, y_i) in one modality would correlate to a change in pixel (x_i, y_i) of the other. Note that some of the sequences in Scene A contain some pedestrians, however they were not labeled due to their relatively low number and low frequency.

Our labeled data provided both the true 2D bounding boxes for all vehicles in the scene present in any image, as well as their corresponding object IDs, which are required for proper object tracking evaluation. In Figure 2.4, we demonstrate some samples from our collected data with the ground truth annotations including objects' bounding box and unique ID for tracking.

We note the parked vehicles present in Scene A (shown in Figure 2.2) are not labeled. This is done to focus on moving vehicles. It is advised to crop the image to exclude the top part of Scene A's samples while training or fine-tuning DL-based models. When using off-the-shelf pre-trained object detectors (*e.g.*, YOLOv3 [149]), detections resulting from these objects can be ignored as done in Chapters 3 and 4.

2.3 Dataset Structure and Statistics

The recordings from each scene are segmented into shorter sequences for more focused analysis. Scene A is divided into 32 sequences, comprising 9,274 images and 6,828 annotations. Scene B, on the other hand, is segmented into 31 sequences with a total of 3,485 images and 3,063 annotations. Consequently, our dataset provides a total of 9,891 vehicle annotations. The discrepancy between the number of images and annotations arises from certain frames lacking any objects. The sequence statistics for each scene, including sequence durations, number of images, events, and objects, are summarized in Table 2.1. On average, each generated sequence is approximately 9 seconds in length, containing around 200 images and 87,000 events. This translates to an average event rate of 10,000 events per second, underscoring the high temporal resolution characteristic of event-based

Table 2.1: Sequence statistics for Scenes A and B in the Dataset. The table details total sequences, duration, number of images, events, objects, and average bounding box area for each scene, providing an overview of the dataset's structure.

Subset	**Total Seqs.**	**Sequence Duration** (s)		**# of Images**		**# of Events**		**# of Objects**		**Average Bounding Box Area** $(pixel^2)$
		Total	Average ± SD	Total	Average ± SD	Total	Average ± SD	Total	Average ± SD	
Scene A	32	397.3	12.42 ±9.94	9274	289.81 ±230.8	2269913	70935 ±59337	6828	213 ±147	1960.5
Scene B	31	147.7	4.76 ±3.55	3485	112.42 ±82.4	3195652	103086 ±31950	3063	99 ±84	4093.2
Total	63	545.0	8.65 ±8.39	12759	202.52 ±194.7	5465565	86755 ±50169	9891	157 ±132	3010.0

Table 2.2: Training and testing split statistics for **Scene A**. This table provides a comprehensive breakdown of the sequences, durations, images, events, objects, and average bounding box areas for the training and testing subsets in Scene A, detailing the dataset's distribution.

	Total Seqs.	**Seq. Duration** (s)	**Total Images**	**Total Events**	**Total Objects**	**Average Bounding Box Area** $(pixel^2)$
			Training			
Average	–	12.2	293.0	74601.0	216.9	1957.4
SD	–	9.8	227.4	63449.8	138.1	924.3
Total	26	316.4	7326	1865024	5423.0	48934.1
%	81%	80%	79%	82%	79%	78%
			Testing			
Average	–	13.5	314.8	64099.0	224.5	2081.1
SD	–	9.6	224.1	29780.6	166.1	1109.4
Total	6	80.9	1889	384594	1347	12486.5
%	19%	20%	20%	17%	20%	20%

sensors. As a result of our labeling, the dataset provides 85 different unique object trajectories in total.

Additionally, we provide sequence-based training and test splits. Training and test splits are critical for the development of any DL-based solutions. We select roughly 80% of the sequences for training and 20% for testing, for both Scene A and Scene B, while ensuring the sequences are well balanced. This is done by carefully verifying a proper 80%-20% (4:1) distribution of the total number of images, events, and objects, in each split. This is demonstrated in Table 2.2 for Scene A, and in Table 2.3 for Scene B. The 80-20 split, a standard heuristic in machine learning, balances the need for substantial training data (80%) with adequate testing data (20%), promoting robust model training and preventing overfitting. This split aims to balance the need for comprehensive learning with the requirement for reliable model validation and generalization to unseen data.

Table 2.3: Training and testing split statistics for **Scene B**. This table offers detailed statistics on sequences, durations, images, events, objects, and average bounding box areas for both training and testing splits in Scene B, illustrating the dataset's balanced division.

	Total Seqs.	**Seq. Duration** (s)	**Total Images**	**Total Events**	**Total Objects**	**Average Bounding Box Area** $(pixel^2)$
			Training			
Average	–	5	109.9	103113.4	96.0	4088.3
SD	–	3	76.4	32434.7	77.1	1123.9
Total	25	116	2747	2577836	2400	102208.1
%	81%	79%	79%	81%	78%	81%
			Testing			
Average	–	5	123.0	102969.3	110.5	4113.5
SD	–	4	97.5	26843.0	99.9	1011.2
Total	6	31	738	617816	663	24681.1
%	19%	21%	21%	19%	22%	19%

A detailed breakdown of each sequence for Scenes A and B is provided in Appendix A (see Tables A.1 and A.2). These breakdowns provide detailed information on each sequence, including the sequence name (identified by the first data timestamp in nanoseconds), duration, number of images, events, objects, and the average area of bounding boxes, along with their allocation to either training or testing splits. This detailed information aids in understanding the dataset composition and its distribution between training and testing.

2.4 Conclusion

In this chapter, we have presented a comprehensive dataset for event-based vision, particularly focusing on object detection and tracking in static scenes. The dataset, featuring vehicle-type objects captured using the DAVIS 240c sensor, fills a significant gap in the realm of event-based vision research. It provides researchers with a tool to explore and develop methodologies in multi-modal object detection and high-temporal-resolution tracking.

Our dataset's primary application lies in the field of object tracking, including high-temporal-resolution tracking, as detailed in Chapters 3 and 4. Here, linear interpolation techniques are employed to enhance method evaluation across various temporal resolutions. Additionally, as shown

in Chapter 6, this dataset serves as a valuable resource for training DL-based object detectors utilizing both event-based and multi-modal approaches. Its design and structured annotations, including object IDs, enable precise and reliable method evaluation, essential for advancing research in this novel field.

However, the dataset is not without limitations. The static nature of the scenes and the absence of ego-motion restrict its application in dynamic scenarios where event-based vision can be more advantageous, such as in HDR and low-light environments. While some pedestrian data is included, it is not the primary focus of this dataset, and their limited number prevents comprehensive labeling. Future iterations of this dataset could expand to incorporate and label more pedestrian data, enhancing its utility and applicability.

In summary, this dataset represents a step forward in the development of event-based vision technologies. Its simplicity and focus make it a practical tool for prototyping and research, particularly in object detection and tracking applications. By providing a foundational resource, this dataset paves the way for future advancements in the rapidly evolving field of event-based vision.

CHAPTER 3

High Temporal Resolution Object Detection and Tracking using Images and Events

Event-based vision is an emerging field of CV that offers unique properties, such as asynchronous visual output, high temporal resolutions, and dependence on brightness changes, to generate data. These properties can enable robust high-temporal-resolution object detection and tracking when combined with frame-based vision. In this chapter, we present a hybrid, high-temporal-resolution object detection and tracking approach that combines learned and classical methods using synchronized images and event data. Off-the-shelf frame-based object detectors are used for initial object detection and classification. Then, event masks, generated per detection, are used to enable inter-frame tracking at varying temporal resolutions using the event data. Detections are associated across time using a simple, low-cost association metric. Moreover, we collect and label a traffic dataset using the hybrid sensor DAVIS 240c. This dataset is utilized for quantitative evaluation using state-of-the-art detection and tracking metrics. We provide ground truth bounding boxes and object IDs for each vehicle annotation. Further, we generate high-temporal-resolution ground truth data to analyze tracking performance at different temporal rates. Our approach shows promising results, with minimal performance deterioration at higher temporal resolutions test (48–84 Hz) when compared with the baseline frame-based performance at 24 Hz.

3.1 Introduction

Object tracking is a common and well-defined task in CV. It entails identifying objects in a scene and tracking their locations across time. The implementations using conventional cameras have been vast and well-established for quite some time [38, 41, 182]. Typically, object trackers utilize an object detection mechanism applied to images, to detect and track present objects across sequential frames based on some association metrics. This results in discrete tracking outputs with rather low temporal resolution, even when the object detection performance is ideal. Such temporal resolutions might be insufficient for high-speed robotics or for applications that require higher tracking temporal resolutions.

Most conventional cameras (hereafter referred to as frame-based cameras) capture images at a relatively low fixed rate of about 30 Hz (or frames per second). Low dynamic range, motion blur, high power consumption, as well as low update rates, are among the main limitations of frame-based cameras.

On the other hand, event-based vision, which is an emerging field of CV, proposes a novel type of bio-inspired sensing modality that offers different physical properties that can be utilized for common CV tasks, including object detection and tracking. These sensors, commonly known as event cameras in the literature, capture per-pixel brightness changes at a very high temporal resolution at the level of microseconds. These brightness changes are referred to as events and are only generated whenever the brightness change of any given pixel exceeds a set threshold. An initial version of this sensor, known as the Dynamic Vision Sensor (DVS), was first introduced in 2008 by Lichtsteiner *et al*. [101].

In general, an event can be defined as:

$$e = \{x, y, t, p\}, \tag{3.1}$$

where x and y denote the 2D-pixel coordinates of the event, whereas t is the timestamp in microseconds of when the event was captured, and p specifies the polarity of the event, which can be

either positive or negative $p \in \{+1, -1\}$, indicating a brightness increase or decrease, respectively.

Unlike frame-based cameras, event cameras generate data asynchronously only at the pixel(s) that undergo a brightness change. These brightness changes (events), other than noise, typically imply motion or highlight changes in the scene. Moreover, event cameras offer numerous advantages compared to standard cameras, including a high dynamic range (HDR) of typically $>$120 dB vs. $\sim$60 dB for standard cameras, no motion blur, low latency (microseconds), high temporal resolution (1 µs per event), and low power consumption [21]. A more in-depth literature survey of this technology can be found in [55].

When it comes to object tracking, the limitations of frame-based cameras can affect performance. Considering their low capture rates, a rapid change in the position or motion of an object being tracked, for example, might not be detected if it occurs at a higher rate than the camera's capture rate. The effects of this might cause undesired outcomes depending on the intended application, as tracking ends up yielding a low temporal resolution output with insufficient data for the inference of other useful characteristics, such as object kinematics (velocity and acceleration rates), or the ability to generate continuous tracking results without the use of data extrapolation techniques.

As for the other frame-based limitations, object tracking can suffer intermittent object detection performance, where objects of interest are not always successfully detected in each frame. This causes some false-negative readings (missed detections) that may result in erratic and inconsistent tracking performance, especially if other means of averaging or filtering are not applied. Theoretically, the maximum achievable tracking rate should be bounded by the camera's synchronous capture rate, generated at discrete times, given an ideal object detection and tracking performance. Alternatively, a high-framerate input source can be used to yield higher tracking resolutions. However, frame-based object detection is computationally expensive and can be very significant in this case, as inference times per frame are usually in the order of several milliseconds, at best using deep learning-based object detectors [18]. This can effectively limit real-time performance, which might be needed given the application. Moreover, consecutive frames might have minimal changes

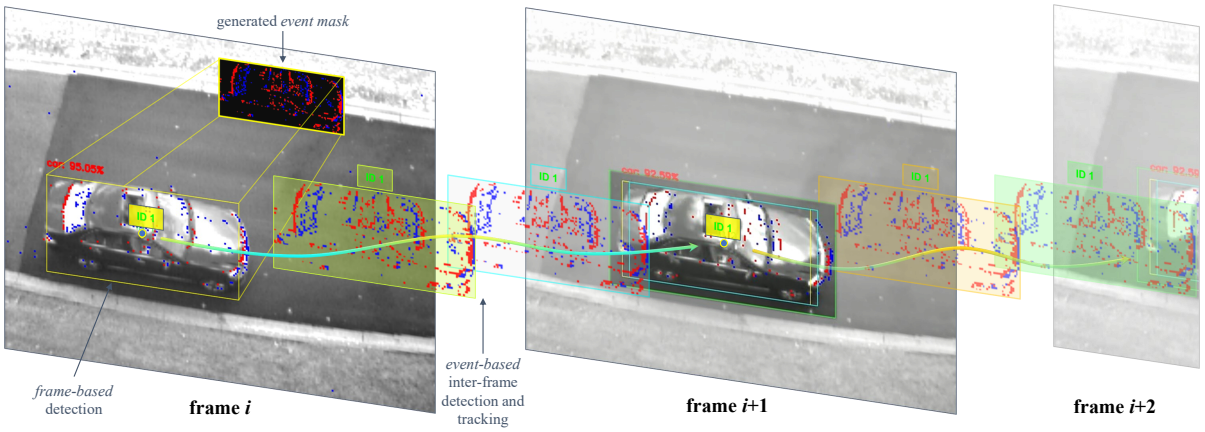


Figure 3.1: A conceptual diagram of our high-temporal-resolution object detection and tracking approach using images and event data. The figure shows three sequential grayscale image frames across time, with events (red and blue dots) overlayed on top, representing their sparse and asynchronous nature. An event mask is extracted whenever an object is detected in a given image, which is then used for inter-frame detection and tracking using events until a new image is captured and the process is repeated.

between them, creating redundant data, yet with the same computational expense per frame. This is in addition to the fact that high-framerate cameras are expensive, require more memory, and consume more power [142].

Nonetheless, event cameras suffer from limitations as well, one of which is the lack of intensity information that regular cameras provide, which causes object classification to be challenging. Although it was shown that intensity images can be reconstructed from events [147], noise and other issues can cause artifacts in the reconstruction. This is evident in scenes with limited changes generated by a camera without any ego-motion applied, in which a significant proportion of the events generated are due to noise. Ego-motion is defined as the 3D motion of a camera relative to the environment [22]. Ego-motion applied on an event camera acts as a trigger that generates events at the edges of the objects within the camera's field of view due to the brightness changes prominent around edge-like features. Accordingly, to achieve more robust detection and tracking, a combined approach would be advantageous.

In this chapter, our main contributions can be described as follows:

1. We present and evaluate a novel hybrid approach to utilize some of the advantages of both

types of sensing modalities (frame-based and event-based vision) to produce higher tracking temporal resolutions. Frame-based vision is used for detecting and classifying objects in a scene (learned approach), whereas event-based vision's asynchronous and high temporal resolution is used for inter-frame tracking by using event masks extracted from the event stream guided by the frame-based detection position (classical approach). Euclidean distance-based object association is used, as the data generated is assumed to be continuous whenever an object is moving, to evaluate the feasibility of higher temporal resolution tracking. Our approach is demonstrated in Figure 3.1.

2. We collect and manually label a multimodal dataset (detailed in Chapter 2) comprising several hours of synchronized image and event data using DAVIS 240c [21]. Our labeled dataset provides both the true 2D bounding boxes for all vehicles in the scene for any image, as well as their corresponding object IDs, which are used for object tracking evaluation[1].

3. To generate matching high-temporal-resolution tracking data for our evaluations, we temporally interpolate our ground truth data multiple times to yield true rates beyond the base framerate of the APS, which is 24 Hz.

4. We assess our approach's performance using state-of-the-art object detection and tracking metrics, at temporal resolutions of 24, 48, 96, 192, and 384 Hz.

3.2 Related Work

3.2.1 Frame-Based Object Tracking

Frame-based multi-object tracking has been well-established in the literature for quite some time. Most works currently utilize direct methods, specifically tracking-by-detection, using optimized object detectors, while focusing on the data association aspect of object tracking [2, 135, 178, 180, 188].

[1]Dataset is available at http://sar-lab.net/event-based-vehicle-detection-and-tracking-dataset/

Recent state-of-the-art trackers, such as DeepSORT [178] and SOTMOT [188], propose different association methods that are performed in an online manner constrained by a trade-off between accuracy and latency. DeepSORT [178], for instance, incorporates motion information based on a recursive Kalman filter [83] and appearance information generated by a pre-trained Convolutional Neural Network (CNN), using Mahalanobis distance, to perform data association on a frame-by-frame basis. Frame-based detections are generated using a fine-tuned FasterRCNN [152]. Meanwhile, SOTMOT [188] employs a one-shot framework based on a modified DLA [183] backbone with multiple parallel branches to perform object detection and data association simultaneously.

Global methods, also known as batch methods, exist as well [135, 180]. However, they are not considered in this chapter due to their limited utility in robotics operating in real-time, as they function in an offline manner. Thus, they require full knowledge of all present and future data for object detection and tracking. Further, it is common for global trackers (and some online ones) to use linear interpolation to cover the gaps in the trajectories of the objects being tracked. On the other hand, some trackers (such as Deep SORT [178]) incorporate motion information to improve data association and mitigate missing detections, using predictions generated by a Kalman filter [83]. Finally, most of these implementations are usually evaluated and compared using common frame-based multi-object tracking benchmarks, such as MOT20 [39], which contains only image frames (no event data).

In our work, we use Euclidean distance [36] as the data association metric. Euclidean distance can be defined as the length of a line connecting any two points. This metric is sufficient for our work, given the modest complexity of the dataset used and the expected continuous nature of the object detection resulting from the added use of event data. Furthermore, in our work, the frame-based-only approach to object tracking is only used as a baseline (at 24 frames per second) to compare with the tracking results of the higher temporal resolutions (48 Hz and above) that use both modalities. Thus, it is irrelevant to include other frame-based approaches in our evaluation. Finally, to constrain the scope of this work, we do not investigate the application of interpolation techniques to fill any gaps in the generated tracking trajectories.

3.2.2 Event-Based Object Tracking

In contrast with frame-based object tracking, event-based object tracking is still in its early stages. In the literature, event-based feature tracking has been the focus of the research community and significant progress has been made. It entails using event data to extract features of different types (*e.g.*, corners) and track them through time [182, 165, 58]. As for event-based object detection and tracking, most works have been application-specific, with few similarities overall [104, 77, 121, 119, 144, 31, 94, 66, 185, 187, 11]. We categorize these works as either event-based or combined (*i.e.*, using images and events) approaches.

A common approach to event-based object tracking is using clustering methods [77, 121, 66, 11]. Clustering is an intuitive approach for event-based object tracking whenever there is no ego motion applied to the camera, thus assuming that events are mainly generated around the moving objects. Therefore, these clusters can track these objects with decent performance. Nevertheless, clustering is less robust against occlusion and can lead to more object ID switching between the objects being tracked.

As for the other, non-clustering, event-based tracking methods, Mitrokhin *et al*. [119] proposed a motion compensation model that enables the detection of objects in a scene by finding inconsistencies in the resulting model and then tracking them using a Kalman filter. They tested their approach on a dataset collected on a moving platform comprising several sequences of varying lighting conditions. The objects were labeled at the time instances of the captured RGB frames. Finally, they evaluated their tracking performance based on a success rate of the percentage of objects detected with at least 50% overlap.

Chen *et al*. [31] proposed an asynchronous tracking-by-detection method for object tracking based on bounding boxes which involved combining events and converting them into frames. Afterward, they used the generated frames with their proposed tracking method and directly compared them with other frame-based approaches. The number of frames generated is dynamic, based on the sum of events captured due to the motion of the objects in the scene. Objects are detected using a contour-based detector and then tracked using an Intersection over Union (IoU) measure

for data association. Finally, they used the same dataset provided in [119] along with Average Precision (AP) and average robustness (AR) metrics for evaluation.

Ramesh *et al*. [144, 143] presented an object tracking method using a local sliding window technique for reliable tracking. Objects are initially detected using a global sliding window to find Region of Interest (ROI) which is only used during the initialization of an object or when the tracking fails to enable real-time performance. Finally, overlap success and center location error metrics were used for quantitative evaluation on a short indoor data sequence [123].

As for the combined approaches using events and frames, the work by Liu *et al*. [104] proposed to utilize the event stream to generate ROIs using cluster-based methods which are then classified by a CNN as either foreground or background. Finally, a particle filter is used to estimate the target's location using the extracted ROIs. This work was mainly meant for detecting and tracking a single object (representing a prey robot); therefore, positional accuracy was used as the evaluation metric.

Zhang *et al*. [185] similarly presented a multi-modal approach to achieve single object tracking. They evaluated success and precision rates on a large-scale dataset annotated at different frequencies, for both vision domains, using a motion capture system. Meanwhile, Zhao *et al*. [187] proposed an object detection method based on color which then tracks a single object using a kernel correlation filter applied to the event data and estimates the distance to the object, while mean Average Precision (mAP) is used to assess the detection performance.

Overall, we noticed that most works in the literature focused on object tracking from a detection perspective, meaning that they only estimated the overall detection and overlap success rates for all objects available. None seems to have evaluated data association performance, which is the common practice in the frame-based domain. This can be attributed to the scarcity of event-based datasets as well as the limitations of the publicly available ones, as most authors emphasized single object tracking and thus did not include ground truth object ID data per annotation. Object IDs are required by the most popular object tracking metrics [38, 39, 111] for evaluating data association performance. In contrast, we provide a fully labeled traffic dataset with bounding boxes and object

IDs for objects of vehicle type. Additionally, to the best of our knowledge, none of the works have explored the use of event data for higher temporal resolution object tracking than the base framerate of a given frame-based camera. Meanwhile, we achieve this here by generating several higher-temporal resolution ground truth data for the acquired sequences, at various rates. These labeled trajectories are then utilized in the evaluation of different approaches for event-based inter-frame frame tracking, using well-defined object-tracking metrics [38, 39, 111]. Accordingly, we assess the feasibility of high-temporal-resolution tracking using a hybrid approach.

3.3 Methodology

In this section, we break down the design of our hybrid approach.

3.3.1 Frame-Based Object Detection

Given temporally synchronized streams of images (frames) and event data, we start with the image stream. A vital first step for tracking objects across time is to detect them when they first appear and in every subsequent frame. As mentioned before, classification using event data alone is challenging; therefore, our approach uses image frames to detect and classify objects wherever they appear in the scene, then tracks them between frames using event data.

To achieve reliable object detection, we utilize two well-known, pre-trained, deep-learning-based object detectors, namely, YOLOv3 [149] and SSD [108], to perform frame-based object detection. These models are used to detect objects in every new image frame, as shown in Figure 3.2, initializing the objects to be tracked and feeding into the Euclidean-based object tracker, described in Section 3.3.3. The frame-based object detectors can be replaced by other frame-based detectors as needed based on the desired minimum ac-curacy and maximum latency requirements. In our work, we use a detection confidence threshold of 50% and a non-maximum suppression threshold of 50% as well for both object detectors used. This process is repeated whenever we read a new image frame.

Figure 3.2: Object detection output on sample images for one of the scenes in our dataset. The object detector used in this figure is YOLOv3 [149]. In this scene, static objects, such as the parked vehicles in the top half of the scene, are disregarded.

3.3.2 Event-Based Object Detection

3.3.2.1 Combining Image and Event Streams Using Window Frames

To make use of an asynchronous event stream, an event-representation method is required. In our work, we accumulate events for a certain interval and incorporate them into a window frame, along with any available image frames. For our application of high-temporal-resolution tracking, the desired tracking rate k must be initially set. k defines the tracking rate our system would utilize to accumulate and parse event data. For example, given that the frames are captured at a rate of 24 Hz, a k value of 48 Hz would indicate that a window frame is collected every 21 ms. The window frame size refers to the duration of the time that the system will read and accumulate synchronized images and event data per window frame. As stated earlier, DAVIS 240c has a frame-based capture rate of 24 Hz; therefore, a new image frame is read around every 42 ms. Thus, using a k value of 48 Hz, every other window frame will contain an image frame (captured by the APS) as well as all the events generated throughout that time (captured by the DVS). This is demonstrated in Figure 3.3. In this work, we experiment with multiple k values, including 24, 48, 96, 192, and 384 Hz, which correspond to window frame sizes of around 42, 21, 10, 5, and 3 ms, respectively.

Accordingly, whenever a window frame containing an image is read, frame-based object detectors output a list of 2D bounding boxes with corresponding object classes for each, as described in Section 3.3.1. Whenever these detections are fed into the object tracker, we generate an event mask

per object detected. These event masks are used to accurately detect and localize the identified objects, using the event data, in the subsequent window frames containing events only (assuming that k is higher than the APS base frame rate). Using the prior example ($k = 48$ Hz), the first window frame would contain an image as well as events, whereas the second would only contain events. Similarly, the third window frame would contain both, while the fourth would contain only events, and so on, as shown in Figure 3.3.

Furthermore, the window frame can either take discrete time steps or use a moving window instead. A discrete step would mean that the window frame would move $1/k$ ms forward for every new frame, as shown in Figure 3.3. Meanwhile, a moving window would incorporate a longer duration of event history for every window frame; thus, some events would be included in multiple consecutive ones. For example, when setting the event-history duration as 50 ms and the tracking rate as 48 Hz, the window frame would read the last 50 ms of event data at any time instant t_i (instead of just 21 ms in the case of discrete time steps), yet it would still move 21 ms forward when loading a new window frame. In general, the window frame would include all of the events available within the time interval $\{t \in \mathbb{R}_+ \mid t_i - 50 \text{ ms} \leq t \leq t_i\}$ at a given time instant t_i. Incorporating a longer temporal history of events can produce higher tracking accuracy, especially at greater tracking rates or resolutions, where larger numbers of event data are accumulated compared to when using a discrete-step window frame. The effects of both parameters, as well as temporally weighting the events, are evaluated later on in this chapter.

3.3.2.2 Event Mask Extraction

As for the event masks, they can be either event-based or edge-based. Event-based masks are produced by extracting all the accumulated events (available in the most recent window frame) that are located within the bounding box of each object detected in the image, as shown in Figure 3.4. Due to the sparse nature of event data, the event-based masks are stored as a sparse matrix of $+1$ and -1 integers, representing the mask's positive and negative events, respectively. Additionally, only the most recent event per pixel is used in the event-based mask's sparse matrix. Moreover, if

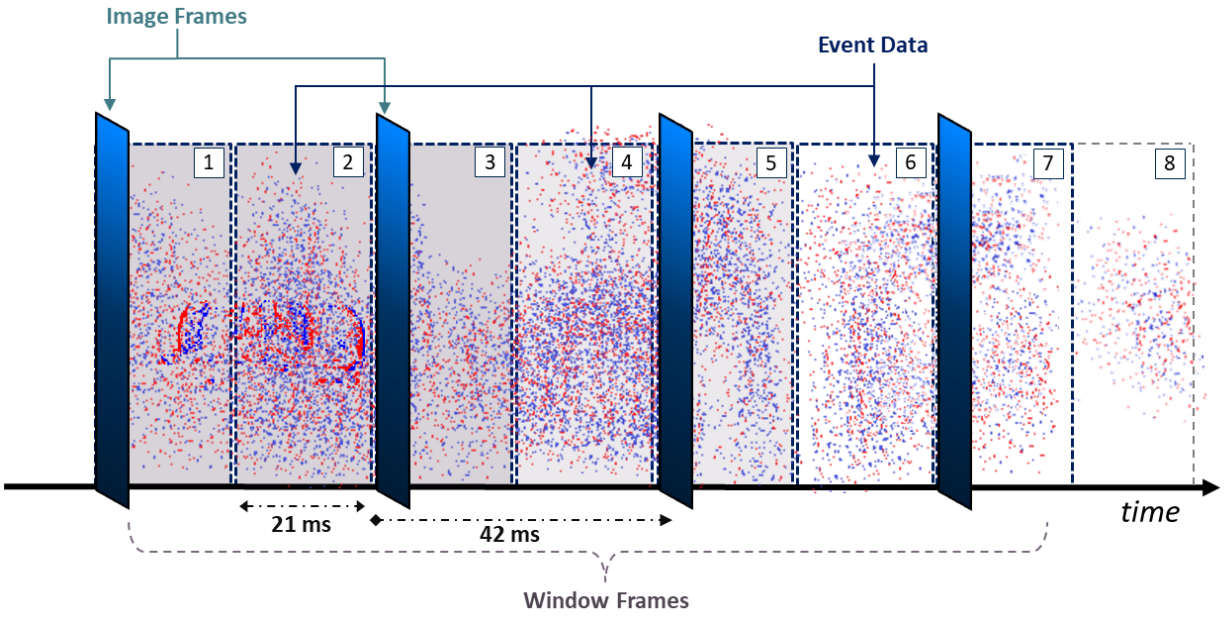


Figure 3.3: Visualization of a synchronized stream of image frames and event data over time. In this example, the image frames are captured at around every 42 ms (at a rate of 24 Hz), whereas the window frame size is set to a temporal resolution of 21 ms (tracking rate of 48 Hz). A window frame encapsulates any image frames and event data available in that specified time frame. A total of 8 window frames are demonstrated in this figure as indicated by their number.

a discrete-step window frame is used, the event mask appends the events found in the next window frame after the object is tracked to improve the tracking robustness in subsequent window frames containing events only. However, this approach assumes that an object is correctly tracked using the event data. Otherwise, if a moving window with a significant amount of event history is used, the event mask is only generated when detecting an object in a given image frame and used without alteration in the subsequent window frames of event data.

On the other hand, edge-based masks are generated using the image's bounding box crop, generated by the frame-based object detector. Given that events are typically generated around the edges of an object whenever there is motion, an edge-based mask can be useful for event-based tracking. To generate an event-based mask, the bounding box crop is initially converted to grayscale (if an RGB image is used), then it is equalized based on its histogram to mitigate low-contrast crops that are either too dark or bright to be able to generate accurate edges. Afterward, an edge-based mask is generated using the Canny Edge Detection algorithm (developed by Canny, J. [25]) which is

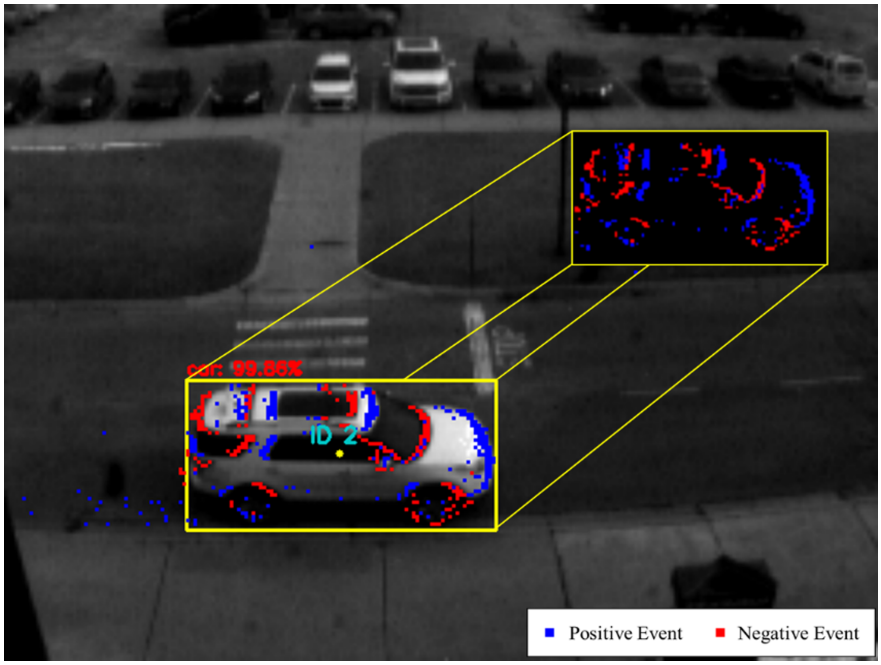
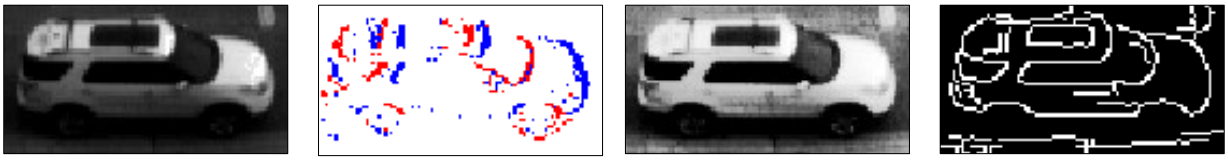


Figure 3.4: The figure demonstrates when an event mask is generated by accumulating the events located within the bounding box, as shown in the top right corner. In this frame, a white SUV is detected, as highlighted by the yellow bounding box (using the frame-based object detector SSD [108]), with 99% confidence. The tracking rate used here is 48 Hz, meaning that the window frame's size is 21 ms and only the events captured during this interval are displayed.

then thresholded to create a binary version of zeros and ones (representing the object's contour). Finally, it is stored in a sparse matrix that represents the event mask of the object. These steps are demonstrated in Figure 3.5. Note that when an edge-based mask is used, the event polarities are no longer utilized. Instead, only the presence of an event at a given pixel is considered.

The motivation behind the edge-based approach is that events are mainly generated at the edges of the objects, as edges represent a sharp intensity change in a given local patch of an image. This way, an edge map would be more robust with respect to tracking an object moving in any direction, whereas, for an event-based mask, events are generated in the direction of motion; therefore, if an object suddenly moves perpendicularly to its prior direction of motion (*e.g.*, vertically instead of horizontally), tracking might momentarily fail until sufficient events are captured and accumulated due to the vertical motion. We can notice this effect on the event-based mask in Figure 3.5(b). The edges around the top and the bottom of the vehicle have almost no events in contrast to the edge-



(**a**) Actual image crop (**b**) Event-based mask (**c**) Histogram-equalized crop (**d**) Edge-based mask

Figure 3.5: Visualization of (a) the actual grayscale image crop based on the bounding box of the detected object; (b) an event-based mask created from the accumulated events in the current window frame used in event-based tracking; (c) a histogram-equalized version of the crop; and (d) the generated edge-based mask used for event-based tracking as well.

based mask in Figure 3.5(d). Nevertheless, the edges of the background are also incorporated into the mask, which might affect the tracking's accuracy and precision. Additionally, the edge-based mask can be affected by poor image conditions, specifically when there is over- or underexposure in the scene.

3.3.2.3 Inter-Frame Object Detection Using Event Data

Once the initial window frame containing an image frame is read, the next window frame is loaded. Assuming a k value of 48 Hz, the second window frame would contain event data only (as demonstrated earlier in Figure 3.3). Therefore, the next step would be to perform event-based object detection and tracking, using the extracted event mask of each object detected in the prior window frame's image. Similar to [144], a search region is used to track an object locally using the available events. Based on the set parameters, the event-based inter-frame object detection and tracking is performed as follows:

1. Create a search region positioned around the center of each of the objects being currently tracked (detected in the latest image). The search region is set 20% larger than the frame-based detection's height and width. Thus, around a 44% larger bounding box size is used in our case (represented by the green bounding boxes in Figure 3.6). This value can be set according to the nature of the objects (expected velocities, etc.). Larger search regions can be used, however, at higher computational costs. Moreover, we add padding to the search region when an object is at the edge of the frame and is exiting the scene, to return a more

accurate object position.

2. Extract all the events (available in the current window frame) located within the search region.

3. For every possible event mask and search region intersection combination:

   (a) Using a sliding window mechanism, create a sparse matrix of the subset of the search region. These events are encoded either spatially or spatiotemporally.

   (b) Perform a cross-correlation between the mask and every search region's subset, as demonstrated in Figure 3.7. This process is mainly a two-dimensional sliding-window matrix multiplication between the event mask and each subset of the search region (starting at the top left corner of the search region). The sum of all the cells, resulting from every matrix multiplication combination, is stored in the corresponding entry of the cost matrix C. The cost matrix C is of size m rows by n columns, which are defined as:

$$m = H_{sr} - H_{em}, \tag{3.2}$$

$$n = W_{sr} - W_{em}, \tag{3.3}$$

   where H_{sr} and W_{sr} are the search region's height and width, while H_{em} and W_{em} are the event mask's height and width, respectively.

4. Based on the highest $C_{i,j}$ entry value, use the best correlating box as the object's inter-frame position. Figure 3.7 shows the best tracking result of this maximum correlation step highlighted in the cyan bounding box, which is the best fit for event-based tracking for the current window frame. Similarly, this is demonstrated in Figure 3.6 by the light-blue bounding boxes. A minimum threshold is typically applied so that the system will only update each object's position if the $C_{i,j}$ value is above a set threshold. This is typically done to avoid updating the object's position based on noise, thus limiting the number of false positives.

(**a**) Event-based mask with a discrete-step moving window.

(**b**) Event-based mask with a temporally-weighted moving window.

(**c**) Edge-based mask with a temporally-weighted moving window.

Figure 3.6: Inter-frame tracking output at 48 Hz in three different modes: (a) event-based mask with a discrete-step moving window with no temporal weighting; (b) event-based mask with temporally weighted events in a 50 ms moving window frame; (c) edge-based mask with temporally weighted events in a 50 ms moving window frame. The inter-frame object position is highlighted by the light-blue bounding box (cyan dot represents its centroid), whereas the yellow bounding box and dot represent the object's position and centroid in the latest image frame, respectively.

5. If successfully detected, update the object's position using the object tracker described in Section 3.3.3. If a discrete-step window frame is used, update the object's event mask by aggregating it with the new event data available within the updated position, assuming the object is correctly detected and that the new events will line up correctly with the previous ones. This step typically improves the tracking robustness, particularly when tracking at very high rates (*e.g.*, >200 Hz), at which fewer events are captured. Otherwise, if a moving window frame is used, the event mask would only update when a new image frame is read.

6. Finally, load the next window frame and repeat the same process according to whether it contains an image or just event data.

Note that when creating the search region (step 3a) to find the object's inter-frame position, we encode the events either spatially or both spatially and temporally. Spatial encoding refers to incorporating the events' x and y coordinates in the tracking process (which is the base case throughout the chapter), whereas temporal encoding incorporates their capture time t as well. Temporal encod-

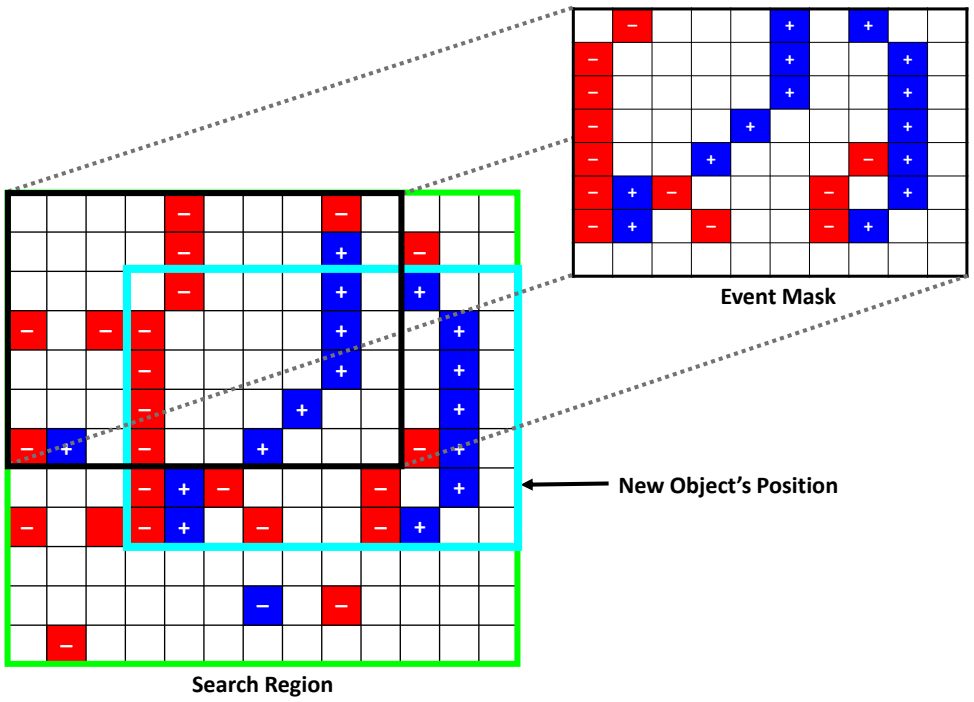


Figure 3.7: Demonstration of the sparse matrix multiplication between the event mask and a sliding section of the search region. This process is used to find the highest correlating position of the object by summing the result of each multiplication, similar to a typical image convolution using a kernel. Based on the results of the sliding window mechanism, the new object's location is set by selecting the highest correlating position as highlighted by the cyan rectangle in this example.

ing is accomplished by weighting the events either equally or temporally. Equal weighting gives all events the same significance, meaning that all events have the same impact on the estimated position of the object. Meanwhile, temporal weighting gives more weight to the most recent events and less weight to the older events. This is visualized in figure 3.8.

To weight the events temporally, we use the following equation for each event:

$$w_{e_i} = \frac{p_{e_i}(t_{e_i} - t_{w_j 0})}{\Delta t_{w_j}} \tag{3.4}$$

where w_{e_i} is the given weight of the event e_i at a specified pixel position; p_{e_i} and t_{e_i} are the polarity and the timestamp of the event e_i, respectively; while $t_{w_j 0}$ and Δt_{w_j} are the window frame j's start time and size (in the same timestamp unit). As described earlier in this section, the window frame size would be equal to either $1/k$ ms, if a discrete-step window is used, or a

Figure 3.8: Demonstration of an image with temporally weighted events (visualized by the transparency effect) overlayed on top. Faded blue and red dots resemble older positive and negative events, respectively. This scene represents the same time instance as the one shown in Figure 3.4 at a tracking rate of 48 Hz, though with an extended 50 ms of event data history compared to 21 ms.

specified duration (longer than $1/k$), if a moving window is used with an extended event history. The resulting weights w_e are appended to the search region's sparse matrix (using the most recent event available at every pixel coordinate) and then used in finding the best object position estimate. In contrast, when the events are weighted equally, the weight w_{e_i} of each event is simply set equal to their defined polarities p_{e_i}. Moreover, the polarity p_{e_i} of any event is set as 1 when using an edge-based event mask to track the objects.

3.3.3 Euclidean-Based Object Tracker

As for the object tracker, we use a simple centroid-based (detections' center x and y coordinates) object tracking algorithm using Euclidean distance [153] as the object association cost across consecutive window frames. Euclidean distance is a metric that is used to find the optimal assignments

to be able to track objects across subsequent frames at any given point with a low computational cost. Moreover, it is appropriate for our application given the continuous nature of the event data and the presumed object detection data, as the centroid of any moving object should be the one closest to its prior center, given that it was successfully detected. The centroid-based tracking algorithm used is based on the work of Adrian Rosebrock [153].

Even though the inter-frame event-based detection (described in Section 3.3.2) fundamentally tracks the objects and estimates their new positions, the detection results are fed into the object tracker to confirm the object assignments. The object tracker uses these detections to either: register new objects with a unique ID, update the positions of the current ones being tracked, or possibly remove the objects that were not successfully matched for n subsequent window frames. Overall, more sophisticated association metrics can be used; however, this work mainly focuses on presenting a novel method to leverage the event data to enable higher-temporal resolution tracking and analyze its feasibility. Thus, the object tracker can be replaced by other tracking-by-detection methods in future studies as desired.

Finally, we summarize our overall object detection and tracking approach in Figure 3.9.

3.4 Experiment Setup

In this section, we describe the dataset that is utilized in the evaluation of our approach, then we define the object detection and tracking evaluation metrics used. Finally, we overview the different tracking configurations applied in our experiment.

3.4.1 Dataset Description

For evaluating our hybrid object detection and tracking method, we utilize the MEVDT dataset previously detailed in Chapter 2. This dataset is captured using the DAVIS 240c sensor [21], a sensor combines a frame-based APS and an event-based DVS, using the same pixel array with a resolution of 240×180 pixels. The APS captures monochrome images at approximately 24 FPS,

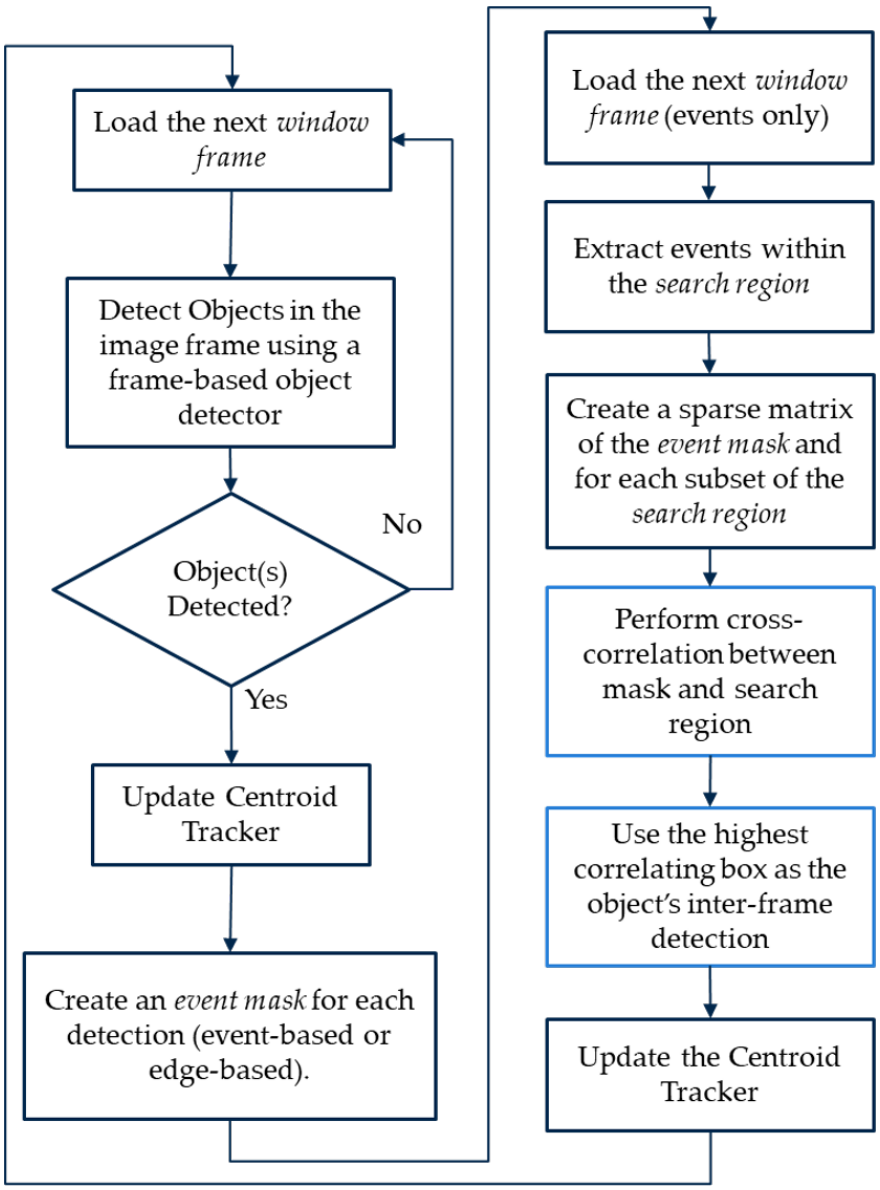


Figure 3.9: Summary flowchart of the overall hybrid object detection and tracking process. The branch on the right would repeat for every consecutive window frame that only contains events given prior frame-based object detections until a new image is read. The window frame size is set before this process starts.

while the DVS captures asynchronous events with a temporal resolution of 1 μs.

We conduct our experiments using data collected from two different scenes, referred to as scenes A and B, as described in Chapter 2. These scenes were selected for their varying proximity of objects to the event camera, affecting the size of objects relative to the frame. The setup involved positioning the static event camera to simulate an infrastructure camera setting, capturing mainly

moving vehicles. Pedestrian data, although collected, are not included in this study due to their sparsity and the challenges in detection caused by their slow movements and distance from the camera.

For quantitative evaluation, we utilize the manually labeled APS-generated intensity images with 2D bounding boxes and object IDs, essential for tracking performance assessment. The dataset comprises approximately 30 sequences per scene, with scene A containing 32 sequences (9274 images, 6828 annotations) and scene B having 31 sequences (3485 images, 3063 annotations), totaling 9891 vehicle annotations.

In our high-temporal-resolution tracking experiments, we require tracking ground truth data at various temporal resolutions beyond the base framerate of 24 Hz provided by the APS. To achieve this, we employ a label interpolation method based on a constant acceleration model. Starting with the annotations at 24 Hz, we linearly interpolate the positions of Bounding Box (BB)s between consecutive frames, maintaining the same object IDs. This process generates ground truth data at higher temporal resolutions of 48 Hz, 96 Hz, 192 Hz, and 384 Hz. The interpolation assumes a linear movement (*i.e.*, a constant acceleration model) between consecutive frames, offering an approximation of object trajectories at these higher temporal resolutions. This approach is particularly important for our experiments, as directly labeling events for tracking at such high temporal resolutions is an extremely challenging task.

For this evaluation, we utilize all sequences from both Scene A and Scene B of the dataset. This comprehensive usage is possible because our approach does not involve training or fine-tuning any DL models. Instead, we apply off-the-shelf pre-trained object detectors, supplemented by an image-processing event-based detection method, to assess the effectiveness of our hybrid object detection and tracking approach across the entire dataset. This allows us to thoroughly evaluate our method's performance under various scenarios and conditions present in the dataset, providing a robust assessment of its capabilities in real-world applications.

3.4.2 Evaluation Metrics

Many evaluation metrics are available to assess detection and tracking performance. In our experiment, we used the novel Higher Order Tracking Accuracy (HOTA) metric, developed by Luiten *et al*. [111], which is used to evaluate multi-object tracking performance. HOTA is particularly useful in assessing the performance of object trackers, as it analyzes the accuracy of the detection, association, and localization of the objects individually and combines them within the same metric. To calculate the final HOTA score, the IoU of localization, detection, and association are calculated. IoU is simply defined as the ratio of the overlap of two detections over their total covered area. The two detections used in the IoU calculation are typically the predicted and the true ground truth detections. As defined by the authors, the foundation of the overall HOTA metric can be described as follows:

- Localization Accuracy (LocA) is the average of all localization IoUs between all possible pairs of matching predicted and true detections of the dataset. Localization refers to the spatial alignment of the predictions compared to the ground truth detections.

- Detection Accuracy (DetA), similar to LocA, measures the alignment between the set of all predicted and ground truth detections. However, it incorporates a defined IoU threshold α to identify which predicted and true detections intersect to find the matching pairs, known as True Positives (TPs). False Positives (FPs) are the predicted detections that do not match, while False Negatives (FNs) are the ground truth detections that do not match. Accordingly, DetA is calculated by dividing the total count of TP over the summation of the count of TPs, FPs, and FNs.

- Association Accuracy (AssA) measures how well a tracker associates detections over time using all object IDs, *i.e*., assesses the whole track of each ground truth object ID using IoUs. For each track, the IoU is calculated by dividing the number of TP matches between the two tracks, divided by the summation of TP, FN, and FP matches between them as well. Ultimately, the AssA is calculated by finding the association IoU over all matching predicted

and ground truth detections.

- The final HOTA value is then generated, using a range of IoU threshold α values to provide one compact value that incorporates the three different components. This value is used to assess the overall object-tracking performance for a specified configuration.

Furthermore, we note that HOTA(0), LocA(0), and HOTA-LocA(0) refer to the same metrics discussed above, though at the lowest α threshold value; thus, localization accuracy does not affect the results. Additionally, DetRe and DetPr refer to the detection recall and precision performance, respectively, whereas AssRe and AssPr refer to the association recall and precision. The recall and precision values can be used to calculate the accuracy values (for both detection and association). Additional details about these metrics can be found in [111].

In addition to the HOTA metrics, we used a subset of the CLEAR MOT [39, 14] metrics, including:

- Mostly Tracked (MT), which is the number of ground truth trajectories that are covered by tracker output for more than 80% of their length;
- Mostly Lost (ML), which is the number of ground truth trajectories that are covered by tracker output for less than 20% of their length;
- Partially tracked (PT), which is the total number of unique ground truth trajectories minus the summation of MT and ML;
- ID-Switches (IDSW), which is the number of ID switches or the number of times a tracked trajectory changed its ground truth one;
- Fragmentations (FRAG), which is the number of times the ground truth trajectory was interrupted or untracked, before resuming later.

According to the authors of these metrics, ID switches are irrelevant when measuring MT, ML, and PT. Therefore, they mostly focus on detection performance for the overall trajectory of each ground

truth object, without considering association accuracy. This can provide some insight into how well an inter-frame event-based object detection system performs. Finally, we note that CLEAR MOT [14] additionally provides relative MT, ML, and PT metrics (sometimes referred to as MTR, MLR, PTR). However, we utilize the absolute variant due to the limited number of unique trajectories in our evaluation dataset (85 in total).

3.4.3 Experimental Parameters and Configurations

To compare and contrast the results of different detection and tracking settings, we evaluated our approach using two frame-based object detectors with three different tracking modes (of varying parameters) for event-based inter-frame object detection and tracking.

The deep-learning, frame-based object detectors used in our evaluation are YOLOv3 [149] and SSD [108]. Both of these pre-trained models provide real-time performance with great accuracy. SSD is more accurate but has higher latency when compared to YOLOv3. Both object detectors are used as is, with the original weights, and without any further fine-tuning or training. Moreover, as mentioned earlier, we set the confidence and the non-maximal suppression thresholds to 50%. Lastly, we only used the 'vehicle' object class, including its different forms (car, truck, bus, etc.), while filtering out the other class types in our evaluation.

As for the inter-frame tracking, applied at higher temporal resolutions above the base rate (24 Hz), we used three modes of different inter-frame tracking parameter combinations:

1. Event-based mask with discrete-step moving window frame with no temporal weighting;
2. Event-based mask with 50 ms moving window frame and temporally weighted events;
3. Edge-based mask with 50 ms moving window frame and temporally weighted events.

These settings were based on the design details presented in Section 3.3 and are shown in Figure 3.6(a)–(c).

To summarize, we evaluate these three different modes with both frame-based object detectors and at the temporal resolutions of 48, 96, 192, and 384 Hz. As for the 24 Hz rate, we only

Table 3.1: Hybrid object detection and tracking results using HOTA metrics (in %) at different temporal resolutions, using the frame-based object detector YOLOv3 [149]. The results are shown for the three different event-based, inter-frame, tracking modes described in Section 3.3. Our approaches represented by modes 2 and 3 show significant promise regarding the ability to leverage event data to generate accurate high-temporal-resolution tracking results.

Object Detector	Tracking Rate	Tracking Mode	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr	LocA	RHOTA	HOTA(0)	LocA(0)	HOTA-LocA(0)
YOLOv3	24 Hz	*	**56.6**	53.0	60.8	54.6	83.6	62.9	87.0	84.2	57.5	68.1	82.0	55.9
	48 Hz	1	53.8	51.2	56.8	52.7	83.5	58.8	86.7	84.1	54.6	65.0	81.9	53.2
		2	**55.4**	**52.9**	**58.4**	**54.5**	83.6	**60.4**	**86.9**	84.2	**56.4**	**66.9**	82.0	**54.9**
		3	54.7	52.3	57.5	53.8	**83.7**	59.5	86.4	**84.3**	55.6	66.0	**82.1**	54.2
	96 Hz	1	49.6	47.1	52.4	49.0	81.6	53.8	86.4	83.8	50.6	60.6	80.9	49.1
		2	**53.6**	**51**.4	**56.2**	**53.5**	82.1	**57.7**	**86.9**	84.1	**54.8**	**65.3**	81.3	**53.1**
		3	52.6	50.5	55.1	52.4	**82.2**	56.6	86.3	**84.2**	53.7	64.1	**81.4**	52.2
	192 Hz	1	44.4	41.6	47.6	43.3	80.9	48.9	85.8	83.8	45.4	54.1	80.9	43.7
		2	**53.2**	**50.5**	**56.2**	**52.9**	**81.2**	**57.7**	**86.7**	84.1	**54.5**	**64.8**	81.2	**52.6**
		3	52.0	49.4	54.9	51.6	81.4	56.4	86.1	**84.2**	53.2	63.3	**81.4**	51.5
	384 Hz	1	36.4	33.1	40.2	34.2	80.9	41.1	85.3	83.8	37.0	44.0	81.0	35.6
		2	**52.5**	**50.1**	**55.3**	**52.6**	80.8	**56.7**	**87.0**	84.1	**53.9**	**63.8**	81.3	**51.9**
		3	51.3	48.9	54.1	51.2	**80.9**	55.5	86.2	**84.2**	52.6	62.3	**81.5**	50.8

* Image-only tracking (excludes event data). Best results per tracking rate and metric are highlighted in **bold**.

used the frame-based object detectors, given that this rate matches the base capture rate of the APS, the results of which were used to set a baseline for the other tracking results and to analyze the feasibility and consistency of incorporating the event data as well to generate high-temporal-resolution tracking results.

Additionally, we formatted our ground truth data for the different temporal resolutions and the resulting tracker outputs in the *MOTChallenge* [38] format, then generated the results using *TrackEval* [82].

3.5 Results and Discussion

Based on the detection and tracking settings specified in Section 3.4, we obtained the results presented in Tables 3.1 and 3.2, using the frame-based object detectors YOLOv3 and SSD, respectively. Moreover, AssA values are plotted against DetA for each temporal resolution (with the resulting HOTA values) in Figure 3.10.

Starting with the baseline frame-based tracking results, at the base image capture rate of 24 Hz, we obtained final HOTA scores of 56.6% and 69% for YOLOv3 and SSD, respectively. This was

Table 3.2: Hybrid object detection and tracking results using HOTA metrics (in %) at different temporal resolutions, using the frame-based object detector SSD [108]. The results are shown for the three different event-based, inter-frame, tracking modes described in Section 3.3. Our approaches represented by Modes 2 and 3 show significant promise regarding the ability to leverage event data to generate accurate high-temporal-resolution tracking results.

Object Detector	Tracking Rate	Tracking Mode	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr	LocA	RHOTA	HOTA(0)	LocA(0)	HOTA-LocA(0)
SSD	24 Hz	*	**69.0**	67.4	70.9	69.7	89.2	73.4	90.1	89.1	70.2	77.2	87.9	67.9
	48 Hz	1	66.6	64.9	68.5	67.0	88.9	70.1	91.1	88.9	67.8	74.9	87.7	65.6
		2	**69.0**	**67.2**	**71.0**	**69.4**	**89.1**	**72.6**	**91.3**	**89.0**	**70.2**	**77.3**	**87.8**	**67.9**
		3	67.6	66.1	69.3	68.2	89.0	70.8	90.9	88.9	68.7	75.9	87.8	66.6
	96 Hz	1	61.0	59.2	63.0	62.1	86.4	64.8	89.6	88.3	62.5	69.4	86.5	60.0
		2	**66.4**	**64.5**	**68.6**	**67.8**	**87.1**	**70.4**	**90.3**	**88.9**	**68.1**	**74.9**	**87.1**	**65.2**
		3	64.4	62.9	66.0	66.0	86.9	67.9	89.8	88.7	66.0	72.9	86.9	63.3
	192 Hz	1	55.0	52.3	58.0	55.0	84.9	59.5	88.6	87.8	56.4	63.0	85.9	54.1
		2	**65.7**	**63.2**	**68.5**	**66.9**	**86.0**	**70.4**	**90.1**	**88.8**	**67.7**	**74.1**	**86.9**	**64.5**
		3	63.3	61.3	65.7	64.8	85.8	67.4	89.7	88.7	65.2	71.7	86.8	62.2
	384 Hz	1	46.3	42.2	50.8	44.0	84.1	52.0	88.0	87.3	47.3	53.2	85.2	45.4
		2	**65.0**	**62.5**	**67.8**	**66.4**	**85.4**	**69.6**	**90.2**	**88.8**	**67.1**	**73.2**	**87.0**	**63.7**
		3	62.5	60.4	64.7	64.2	85.2	66.4	89.8	88.7	64.4	70.6	86.9	61.3

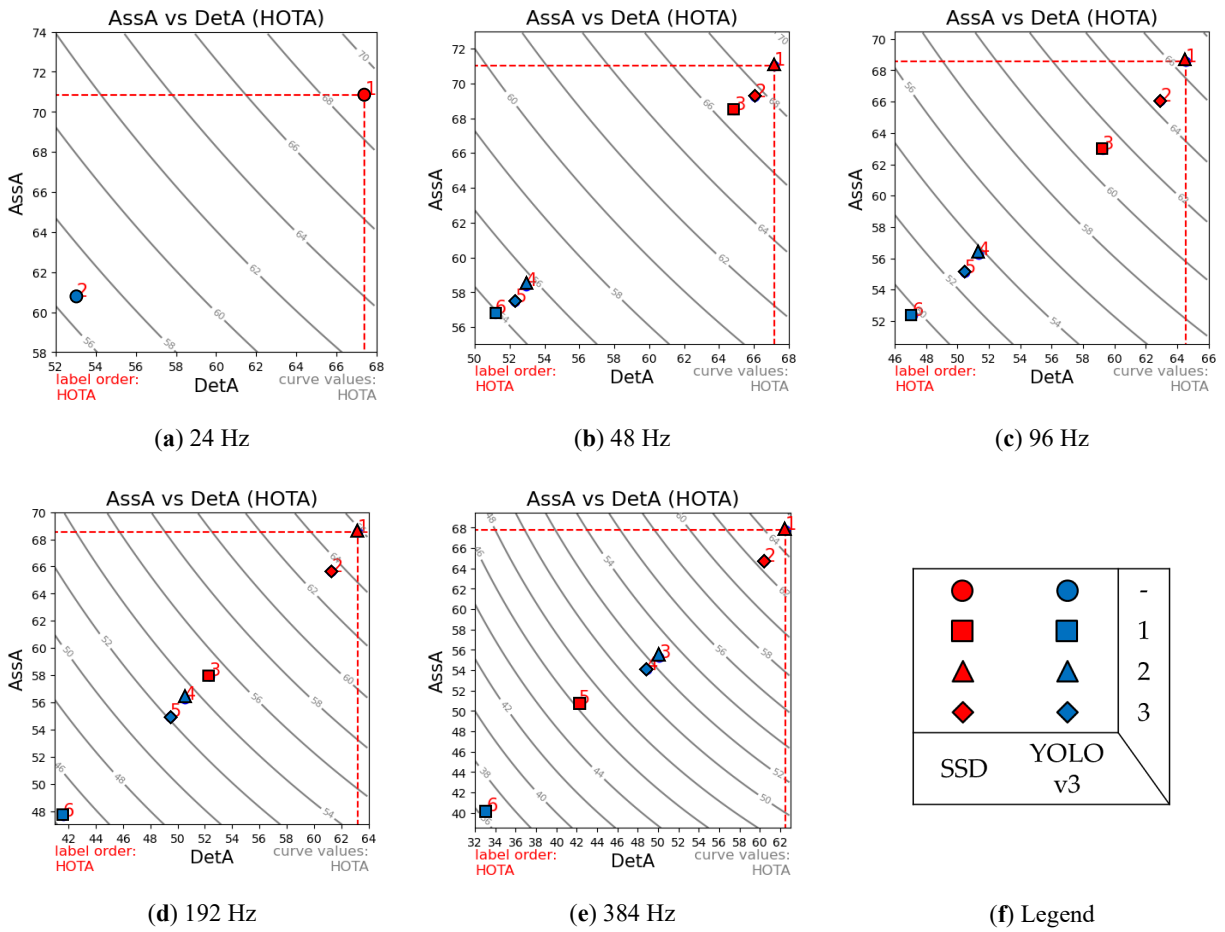
* Image-only tracking (excludes event data). Best results per tracking rate and metric are highlighted in **bold**.

expected given that SSD is a more accurate object detector, as is highlighted by its DetA of 67.4% compared with 53.0% for YOLOv3. These values were used as the baseline values to compare our three different event-based inter-frame object tracking approaches at various temporal resolutions.

Applying the approach specified by Mode 1, which used event-based masks without history or temporal weighting, we noticed that the outcomes of most HOTA metrics significantly deteriorated with higher temporal resolutions. This is the result of a lower number of events being available to track with smaller window frame lengths. A tracking rate such as 384 Hz has a temporal interval of only 2.6 ms.

On the other hand, Mode 2, which also used an event-based mask but with a temporally weighted event history of 50 ms, consistently yielded the best performance when using either frame-based object detector. Mode 3, which used an edge-based mask instead, slightly underperformed Mode 2 but provided similar consistency.

Overall, the approaches represented by Modes 2 and 3 proved that high-temporal-resolution tracking is possible by incorporating event data without any significant impact on performance. In Mode 2's configuration, the HOTA values deteriorated slightly, declining from 56.6% and 69.0%



(**a**) 24 Hz (**b**) 48 Hz (**c**) 96 Hz

(**d**) 192 Hz (**e**) 384 Hz (**f**) Legend

Figure 3.10: Comparison between the results of the different tracking configurations for various temporal resolutions. AssA is plotted against DetA with the resulting HOTA values marked for tracker configuration, for temporal resolutions of (**a**) 24 Hz - (**e**) 384 Hz. The legend (**f**) defines the symbols used according to the object detector used and tracking mode. Results show a linear correlation between the AssA and DetA, with Mode 2's approach outperforming the other configurations for either object detector.

(when using YOLOv3 and SSD at 24 Hz) to 52.5% and 65.0%. This translates to a relative performance deterioration of just 7.24% and 5.8%, for YOLOv3 and SSD, respectively.

Similarly, Table 3.3 shows the results of the selected CLEAR MOT [14] metrics for every tracking configuration. Consistent with the previous results, Mode 2's configuration shows very minimal deterioration in tracking performance. As for the SSD-based configuration, the baseline tracking of 24 Hz had an MT of 40 and a PT of 45 with no ML objects, which was minimally affected by the higher temporal resolutions, as shown in the results for the highest rate of 384 Hz

Table 3.3: Hybrid object detection and tracking results using a subset of CLEAR MOT metrics [14] for different tracking configurations and temporal resolutions. The selected metrics provide extra insight into the behavior and the quality of each tracking configuration. Mode 2's tracking configuration consistently outperformed the others in all metrics, deteriorating slightly with increasing temporal resolutions.

Tracking Rate	Tracking Mode	YOLOv3					SSD				
		MT	**PT**	**ML**	**IDSW**	**FRAG**	**MT**	**PT**	**ML**	**IDSW**	**FRAG**
24 Hz	*	**27**	52	6	18	21	**40**	45	0	16	**33**
48 Hz	1	21	58	6	30	76	39	46	0	29	70
	2	**27**	52	6	30	**22**	**40**	45	0	29	**34**
	3	25	54	6	30	23	**40**	45	0	29	35
96 Hz	1	15	62	8	50	550	25	59	1	48	749
	2	**25**	53	**7**	50	**502**	**37**	47	1	48	**715**
	3	23	55	**7**	50	517	36	48	1	48	727
192 Hz	1	8	66	11	54	635	16	65	4	49	908
	2	**25**	52	**8**	**50**	**505**	**37**	46	**2**	**48**	**715**
	3	23	54	**8**	**50**	526	35	48	**2**	**48**	738
384 Hz	1	3	69	13	86	695	9	68	8	67	1054
	2	**25**	52	**8**	**75**	**507**	**37**	46	**2**	**61**	**721**
	3	22	55	**8**	**75**	532	35	48	**2**	62	742

* Image-only tracking (excludes event data). Best results per tracking rate and metric are highlighted in **bold**.

with MT, PT, and ML of 37, 46 and 2, respectively. Additionally, the YOLOv3-based configuration had MT, PT, and ML baseline tracking results of 27, 52, and 6, respectively, which insignificantly declined at the tracking resolution of 384 Hz, with only two fewer objects MT that became ML instead. Similarly, Mode 3's configuration was a close second at varying tracking resolutions. Meanwhile, Mode 1's configuration performed progressively worse with higher temporal results. We note that there are a total of 85 unique object trajectories in the whole dataset, as shown in Table B.1. Therefore, MT, PT, and ML always add up to a total of 85. As expected, IDSW got marginally worse with increasing rates for each of the three modes, whereas FRAG suddenly increased at the temporal resolution of 96 Hz, then stabilized, except for Mode 1, which continued to worsen at increasing rates. The total number of ground truth detections for each rate is also provided in Table B.1 for reference.

In general, the results show that temporal weighting of events is vital when using event-based

data. Temporal information is a valuable component of asynchronous events which synchronous, fixed-rate, images lack. Our first approach, represented by Mode 1, confirms this hypothesis, where the tracking performance was significantly affected by increasing temporal resolutions, regardless of the frame-based object detector used. As for the third approach, used in Mode 3, edge-based masks were heavily dependent on the captured image quality. Given the limitations of frame-based cameras, this constrains the performance of event-based vision in challenging scenes, making the system less robust given its low dynamic range and capture rates. In our evaluation, event-based masks proved to be more robust, with lower computational costs.

3.6 Conclusions

In this chapter, we have presented a novel way of using frame-based and event-based vision data to enable high-temporal-resolution object detection and tracking. We leveraged state-of-the-art frame-based object detectors to initialize tracking by detecting and classifying objects in a scene using synchronous image frames, then generated high-temporal-resolution inter-frame tracking using event data. We developed and compared three different approaches for event-based detection and tracking and analyzed their performances at several temporal resolutions. Moreover, we used a simple and low-cost association metric, that is, Euclidean distance, to match object detections across time.

We evaluated these approaches using our dataset for two traffic scenes, obtained using a static camera with no ego-motion applied. We collect the data using the DAVIS 240c, which combines a frame-based and an event-based sensor using the same lens, generating synchronized image and event data streams. Furthermore, we manually labeled all the vehicles within the scene with accurate bounding boxes and an object ID for every trajectory, using the images generated by the frame-based camera. Then, we generated high-temporal-resolution ground truth trajectories, for object detection and tracking, by temporally interpolating the labeled data, for the tracking rates of 48, 96, 192, and 384 Hz. Finally, we evaluated the results of our different approaches and

corresponding configurations using HOTA and a select few CLEAR MOT metrics.

Our results show that out of the three methods presented, event-based masks, combined with temporal weighting of events and a sufficient temporal history, yielded the most consistent performance with minimal deterioration as we progressively increased the tracking rates and the corresponding temporal resolutions when compared with the baseline frame-based performance at 24 Hz. Moreover, edge-based masks with temporal weighting showed promise as well, ranking very close to the prior approach, whereas our first approach, using event-based masks but without temporal weighting, resulted in the worst performance with the most degradation as we increased the temporal resolutions.

In conclusion, our work shows that a hybrid approach that leverages both image and event data to generate higher tracking temporal resolutions is feasible, with very consistent performance. Our labeled dataset provides a quantitative means of assessing different event-based tracking approaches, which we hope will encourage the production of other challenging labeled event-based datasets for object tracking in the future, given that the presented dataset might not provide the most challenging scenarios that would require more sophisticated detection and tracking approaches. This can be attributed to the relatively low number of available object occlusions and objects present in the scene at any given instant, as well as the limited resolution of the event-based sensor used. Moreover, when considering tracking different object types, we note that classical approaches might not be ideal for objects of dynamic shapes that change at very rapid rates.

This work opens doors for future research, such as into the use of more advanced association metrics tailored for both of these sensing modalities, a more dynamic approach that is less dependent on either, or the exploration of a fully event-based approach for the entire object detection and tracking process.

CHAPTER 4

Improving High Temporal Resolution Event-Based Vehicle Detection and Tracking

Event-based vision has been rapidly growing in recent years justified by its unique characteristics such as its high temporal resolutions ($\sim$ 1 us), HDR (>120dB), and output latency of only a few microseconds. This chapter further explores a hybrid, multi-modal, approach for object detection and tracking that leverages state-of-the-art frame-based detectors complemented by hand-crafted event-based methods to improve the overall tracking performance with minimal computational overhead. The methods presented include event-based BB refinement that improves the precision of the resulting BBs, as well as a continuous event-based object detection method, to recover missed detections and generate inter-frame detections that enable a high-temporal-resolution tracking output. The advantages of these methods are quantitatively verified by an ablation study using the higher order tracking accuracy (HOTA) metric. Results show significant performance gains resembled by an improvement in the HOTA from 56.6%, using only frames, to 64.1% and 64.9%, for the event and edge-based mask configurations combined with the two methods proposed, at the baseline framerate of 24 Hz. Likewise, incorporating these methods with the same configurations has improved HOTA from 52.5% to 63.1%, and from 51.3% to 60.2% at the high-temporal-resolution tracking rate of 384 Hz. Finally, a validation experiment is conducted to analyze the real-world single-object tracking performance using high-speed LiDAR. Empirical evidence shows that our approaches provide significant advantages compared to using frame-based object detectors at the baseline framerate of 24 Hz and higher tracking rates of up to 500 Hz.

4.1 Introduction

In the last couple of years, the neuromorphic event-based vision has been gaining attention in the literature and growing exponentially [12, 55]. Event-based sensors, first introduced in 2008 [101], propose a novel type of sensing modality with distinct and advantageous characteristics compared to the typical frame-based cameras. These sensors, commonly referred to as event cameras, capture brightness changes asynchronously and independently per each pixel of the sensor's pixel array. Each of these captured brightness changes is known as an event. Every event consists of 4 different types of information including a microsecond-resolution timestamp t of when it was detected, an x and y pixel coordinates at which the event has occurred, and a polarity p indicating the type of brightness change that was registered (*i.e.*, positive or negative). Accordingly, an event is defined as $e = \{t, x, y, p\}$.

In contrast, conventional frame-based cameras capture images synchronously at a fixed rate (typically $\sim$30 FPS), recording the color intensity of each pixel, regardless of whether there were any changes, in every frame generated at a fixed sampling rate. This causes frame-based cameras very susceptible to producing redundant data that may resemble a static background in a given scene, especially when the camera is stationary (e.g. undergoing limited, to no, motion). Meanwhile, event cameras would mostly capture changes in the scene, often resembling motion, at the instances of their occurrence. Nevertheless, event cameras can be less effective in scenes of limited to minimal motion, where there would be a lack of visual signal to reliably utilize this modality on its own. Thus, causing the unimodal event-based implementations to possibly be unreliable in some scenarios. Overall, we illustrate the difference between the visual data output of the two modalities (*i.e.*, frame-based and event-based) in Figure 4.1.

The main specifications of a typical event camera include a very high temporal resolution (1 µs per event making it robust to motion blur), low latency in the order of microseconds, and a high dynamic range (HDR) of over 120 dB (compared to $\sim$60 dB of conventional frame-based cameras) while requiring considerably less power [55, 21]. Given these properties, event-based vision proposes an exciting domain with great promise if explored and applied properly. Current

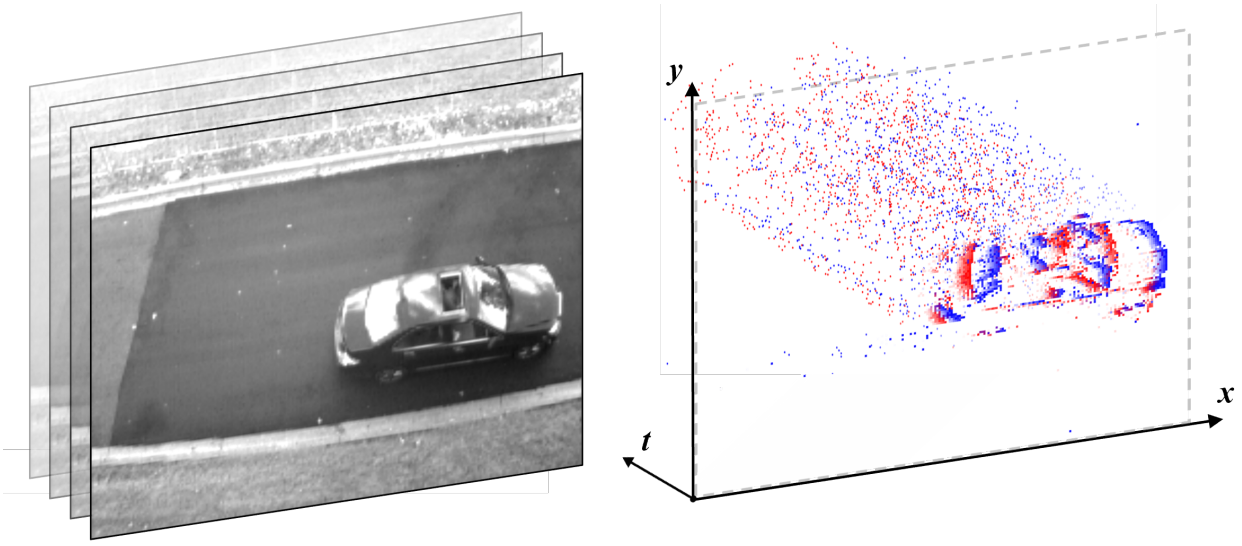


Figure 4.1: Comparison between the frame-based and event-based modality output, resembled by (**a**) the synchronous images captured by a frame-based camera, and (**b**) the asynchronous events captured by an event-based sensor. Notice how the events are mainly generated around the edges of the moving vehicle, where brightness changes exist), in contrast with the images captured by the frame-based camera where the static background is redundantly captured across consecutive frames at a fixed sampling rate.

works have utilized these specifications for different applications such as motion deblurring [80], high-framerate HDR video synthesis [148], image reconstruction from events [147], and enhanced object detection [95] to name a few.

While the potential behind this novel visual-sensing technology is evident, we believe that it can provide optimal benefits when incorporated with frame-based vision, as both modalities can be complementary to each other when they are correctly utilized. Such an approach can enable a more robust perception performance for different automated applications.

In this chapter, we explore a combined frame- and event-based approach for vital computer vision tasks, namely object detection and tracking. Object detection is an essential component for automated systems to provide awareness of the surroundings at a given instant. Meanwhile, object tracking enables the system to associate these detections across time, supplying the temporal element for its interpretation of its surroundings. Both components, although challenging and processing-intensive, are critical for a complete and reliable system perception performance, as

they assist in different tasks, such as motion planning and obstacle avoidance, and play an important role in various applications in robotics, including traffic monitoring and surveillance systems [76], and autonomous vehicles [146, 29].

Object detection performance can vary based on the method used. Deep Neural Network (DNN)-based object detectors have recently dominated the state-of-the-art [149, 108, 151, 102, 152], thanks to the unparalleled advancements in deep learning, in general, [92], and the emergence of deep CNNs specifically [89, 160]. Nevertheless, the different implementations in the literature are often constrained by a trade-off between object detection accuracy and latency. YOLOv3 [149], for instance, offers real-time inference speeds, however, at the cost of lower accuracy and possibly more inconsistent performance, which could affect the overall object tracking performance due to the intermittent detections. On the other hand, FasterRCNN [152] offers better object detection accuracy which the object tracking framework can benefit from, however, at the expense of considerably higher latency, making it less ideal for real-time detection and tracking systems.

Similarly, the temporal resolutions of frame-based object detection and tracking can be limited by the framerate of the input source which is typically fixed, such as a camera that typically has a low output framerate. This sets an upper bound for the resolutions of object tracking, given that interpolation techniques are excluded which are not beneficial for a real-time online system. Furthermore, even if a higher framerate source is used along with a DNN-based object detector, the system's operational latencies would be further impacted, imposing stringent hardware requirements to be able to achieve real-time computation performance.

In this chapter, we extend and improve on our prior work [45] presented in Chapter 3, which explored the feasibility of high-temporal-resolution object detection and tracking using a hybrid multi-modal approach that incorporates synchronized image and event data, by presenting two additional methods that improve the overall object detection and tracking performance using event-based techniques. First, we improve the precision of BBs proposed by frame-based object detectors using a combination of event data and classical computer vision methods. Second, we enhance the robustness and consistency of frame-based object detectors using event-based detection methods.

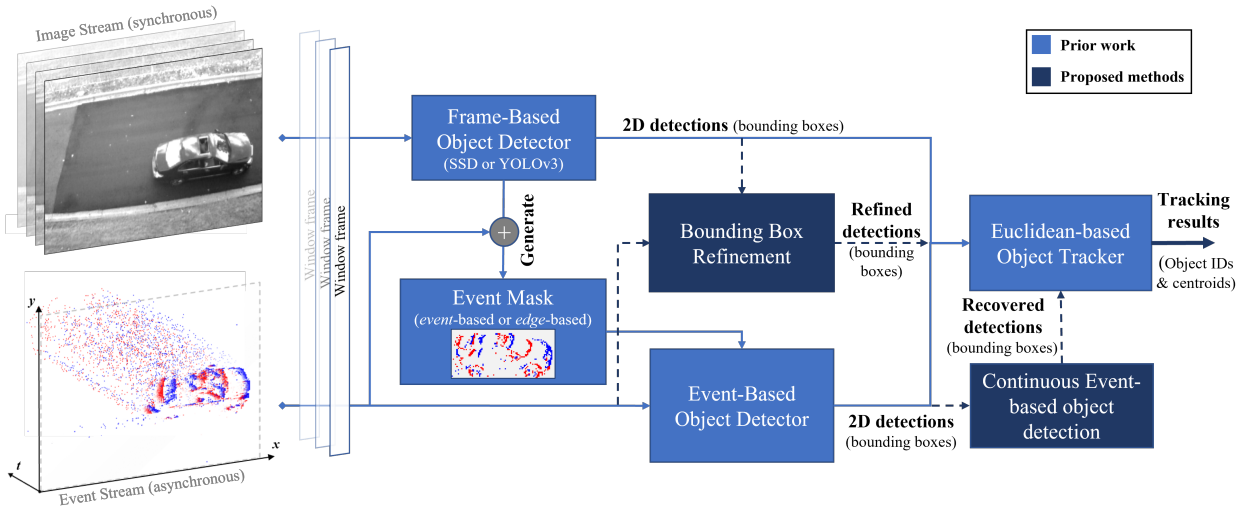


Figure 4.2: Block diagram demonstrating the hybrid multi-modal object detection and tracking framework with the proposed methods, which include BB refinement and continuous object detection using event-based methods. Visual streams from both modalities are synchronized and passed through the framework to yield multi-object tracking results.

This method is automatically initiated whenever the frame-based object detector fails to detect an object in a given frame, thus improving the object detection reliability and the corresponding tracking performance by leveraging the high-temporal-resolution event data. Third, we numerically assess the effects of these methods under different frame-based object detection models using a fully labeled dataset (at multiple tracking rates) along with state-of-the-art Multi-Object Tracking (MOT) metrics, including the higher order tracking accuracy (HOTA) metric for evaluating multi-object tracking. This is followed by a simple computational cost analysis for the presented event-based methods in comparison with the frame-based components. Finally, we validate our work with a real-world experiment using a high-speed LiDAR to provide high-temporal-resolution positional measurements of a vehicle being tracked. The general framework of this work along with the proposed methods is demonstrated in Figure 4.2.

Overall, the main contributions of this chapter are summarized as follows:

- We present an event-based BB refinement method for static scenes, and an event-based method for recovering undetected objects in the frame domain, which improves the object detection and tracking performance compared to the frame-based baseline.

- We conduct an ablation study that quantitively verifies the benefits of each introduced event-based method and their optimal combination using the HOTA metric is presented.
- We provide a computational latency analysis for the introduced methods as well as the core components of the proposed system.
- We perform a real-world validation experiment using a high-speed LiDAR that evaluates how well the presented framework, including the additional event-based methods, estimates the vehicle position at different temporal resolutions and tracking rates.

4.2 Related Work

4.2.1 Frame-based Approaches

Frame-based object detection and tracking methods have had significant developments throughout the last decade. The advancements of DNN-based object detectors have enabled a very robust object detection performance [149, 108, 151, 102, 152], which is complemented by various data association techniques to achieve multi-object tracking (MOT) [188, 178]. This approach of object tracking is commonly referred to as tracking-by-detection.

When it comes to object detection, deep learning-based object detectors are categorized as either one-stage [149, 108, 151, 102] or two-stage detectors [152]. While two-stage detectors are often more accurate and robust, one-stage detectors sacrifice some accuracy for inference speeds making them more appropriate for real-time systems but with degraded performance. One-stage object detectors include YOLOv3 [149], SSD [108], RRC [151], and RetinaNet [102], whereas two-stage object detectors mainly include FasterRCNN [152].

As for object tracking, State-of-the-art MOT implementations are designed to operate in either an online or a global manner. Online methods are more appropriate for real-time robotic systems compared to global methods [135, 180] which require full knowledge of all the current and future data (more suitable for other types of applications). The output of these implementations is typ-

ically evaluated using well-known object tracking metrics such as the CLEAR MOT [14] or the more recent HOTA [111] metrics on various MOT benchmarks such as MOT20 [39].

Analogous to the literature, we leverage a tracking-by-detection approach to enable multi-modal object detection and tracking using image frames as well as events. Focusing on the object detection aspect, our combined approach accomplishes this by the use of pre-trained, one-stage, frame-based object detectors, specifically YOLOv3 [149] and SSD [108] while employing a simple data association metric, which is Euclidean distance [36]. The choice of this data association metric is based on the assumption that the resulting detections, ideally, should be continuous given the high-temporal-resolution nature of event data. Further, using methods discussed later in this chapter, frame-based object detections are used as the basis for enabling the detection and tracking of the objects in the scene in between frames at varying temporal resolutions, using the event data.

4.2.2 Event-based Approaches

On the other hand, event-based object detection and tracking methods, compared to the frame-based implementations, are in their preliminary stages and have yet to utilize the event data to its full potential. Unlike the frame-based domain, event-based vision has had varying approaches when it comes to either object detection or object tracking. For instance, single-modal event-based object detection is currently in the experimental phase, where learned methods make up the majority of the state-of-the-art. Learned implementations typically combine and embed events in an image-link representation which are used with modified frame-based DNN architectures to make them compatible with event data, in either a temporal [96, 136, 24] or a non-temporal [78] manner. Recurrent and temporal approaches are typically more suitable for event data, given that events only provide brightness changes and not absolute brightness, at a given point in time, in contrast to frames. Thus, the temporal approaches would incorporate meaningful input history, instead of treating each split, of the input stream, independently, yet at the expense of higher computation. Although the single-modal event-based approaches are promising, their performance typically lags behind both the frame-based solutions, under normal conditions, as well as the combined solutions

that incorporate both modalities [95, 96, 167, 32, 105]. The main reason behind this is due to the nature of the event data, especially in scenes where there is limited motion. Besides, the lack of enough labeled datasets, available for training deep learning models, only increases the performance gap between both domains. In our work, we present a multi-modal approach that leverages pre-trained frame-based object detectors to initiate objects in the scene. Afterward, these detections are used to generate templates for each object, which we refer to as event masks, in order to detect objects at varying tracking rates using the asynchronous and high temporal resolution event data in a temporal manner. Moreover, unlike fully learned approaches, a mixed approach of learned and classical methods can be more computationally efficient when requiring very high tracking rates. Furthermore, a classical and designed detection method does not require large datasets of labeled data for implementation, which is still a constraint in the event-based vision domain.

As for event-based object tracking, most works focus mainly on the detection aspect with typically a single-object tracking approach. One common approach is the use of clustering techniques [77, 121, 11, 66]. Clustering events is a low-cost and intuitive way for simple object tracking applications enabled by the nature of events that often resemble movement, as they would mainly be generated around the objects that are in motion. Nevertheless, clustering can be prone to object collisions and does not apply to object classification. Conversely, non-clustering methods have explored various single and multi-modal object detection and tracking techniques [185, 187, 104, 31, 144]. For example, Chen *et al*. [31] proposed an event-to-frame conversion algorithm to enable an asynchronous tracking-by-detection approach. Meanwhile, Ramesh *et al*. [144] presented an online object tracking framework with a moving event camera using a local sliding window technique, with a global object re-identification using an event-based object detector whenever the tracker loses the object.

Overall, we notice that few event-based object tracking works focus on real-world objects such as vehicles, where most use data of shapes in indoor scenes, and approach this problem from a single object tracking aspect (*i.e*., without the use of well-defined MOT metrics like in the frame-based domain). A primary constraint behind this is the limited number of labeled object detection

and tracking event-based datasets available publicly. Our prior work [45] provided the first fully labeled event-based dataset conforming to the typical MOT standards [38], including object IDs, unique to each object's trajectory, along with 2D BBs provided at multiple temporal resolutions and tracking rates.

To summarize, our previous work [45] focused on exploiting the high-temporal resolution of the event data to enable high-temporal-resolution object detection and tracking using a combined approach that utilizes learned frame-based object detectors and classical event-based methods. Our proposed framework was evaluated using a labeled vehicle dataset with state-of-the-art MOT metrics [111]. However, our work was constrained by the performance of the frame-based detectors to initiate detections and redetect objects at every new image frame. In this work, we propose two methods that help boost the detection performance, where we use event-based techniques to improve the accuracy of the generated BBs and to support object detection when a frame-based object detector fails to detect an object, previously tracked, in a given image.

4.3 Methodology

In this section, we initially summarize the hybrid framework presented in our prior work [45] in more detail, then describe some of its limitations and the motivation behind the additional methods introduced in this chapter. Afterward, we present and break down the proposed event-based methods to improve object detection and tracking. Finally, we perform an ablation study to assess the effectiveness of these methods and figure out the optimal combination to use.

4.3.1 High-Temporal-Resolution Object Detection and Tracking Framework

In our prior work [45], we have introduced a hybrid approach, that leverages both frame-based and event-based vision modalities to enable high-temporal-resolution object detection and tracking (as demonstrated in Figure 4.2). We break down the framework in the following subsections.

4.3.1.1 Parsing the multi-modal streams using window frames

To incorporate the data streams of both modalities, the temporally synchronized image and event streams are divided into a moving window frame containing any available image frames as well as a predefined interval of event data (e.g., 50 ms). Based on the desired tracking rate of k Hz, the window frame will move forward $\frac{1}{k}$ ms for every step, with an interval size ΔT. For example, given a tracking rate k of 200 Hz and ΔT set as 50 ms, the window frame will take steps of 5 ms while containing the latest 50 ms of event data. Note that inter-frame event-based object detection is executed only if the desired tracking rate k is set higher than the framerate of the images (ideally $\leq \frac{k}{2}$ Hz).

4.3.1.2 General multi-modal detection and tracking framework

Following a tracking-by-detection approach, pre-trained frame-based object detectors (*i.e.*, YOLOv3 and SSD) are used to detect objects in the image domain (when available in the frame) and initiate their tracking. Meanwhile, an event-based, inter-frame object detection method is used to detect previously detected objects in the event domain, in the blind time between consecutive frames. Accordingly, whenever a window frame containing an image is read, the frame-based detections are used to initiate the object tracker with these objects by associating each with a unique object ID. At the same time, the resulting two-dimensional BBs of the detected objects, representing their location relative to the frame, are used to generate templates that are later required for the event-based object detection process. We refer to these templates as event masks. Note that the event masks are of the same size as the objects detected in the image frames. Afterward, these event masks are used to enable inter-frame object detection in the subsequent window frames that only contain events, until a window frame containing an image is read at which point this process is repeated. Throughout this process, the detection results of each window frame are fed into the object tracker, which associates the latest detections by initiating any new objects that entered the scene, updating the position of previously tracked ones, and removing the objects that have left the scene. Moreover, we utilize a simple Euclidean distance [36] minimization function to asso-

ciate recent detections with the objects currently being tracked. This simple association metric was chosen to give more emphasis on the cooperative multi-modal object detection process and is motivated by the expected continuous tracking results due to the high temporal resolution of the event data.

4.3.1.3 Event-based object detection using event masks

Here, we break down the event-based object detection process referred to in our framework. Given a set of currently tracked objects with corresponding event masks, the event-based object detection is achieved using a sliding window mechanism of matrix multiplication between each object's event mask and the events that are available in the current window frame and located within the confined search region. The search region is generated around the latest detected position of each object, albeit with a larger area to cover all possible displaced positions. The results of all the possible combinations of this sliding window mechanism are stored in a cost matrix. Afterward, the highest value in the cost matrix, which resembles the best correlating position, is used as the object's inter-frame position, as long as that value meets or exceeds the preset score threshold. Otherwise, no detection is generated for the corresponding object within that window frame. Therefore, this results in a missed inter-frame detection which creates gaps in the estimated trajectories of the tracked objects.

Note that the event-based object detection process is only applied to the objects that were previously detected in the most recent window frame that contained an image using the selected frame-based object detector. Thus, if an object was not detected in the latest image in the data stream, the event-based object detection process is not initiated until the object is detected in a future image that precedes a window frame of events. Further, we note that the events in each window frame are temporally weighted, giving the highest weight to the newest events and lower weights for older events. By giving more significance to the newest events, which typically resemble the latest movements of the object, this process was found to improve event-based object detection precision[20] and is the applied practice throughout this chapter.

We have also introduced and evaluated two different types of event masks (*i.e.*, event-based and edge-based) which are demonstrated in Figure 4.3. Event-based masks (shown in Figure 4.3(a)) are generated by accumulating all of the events located within the 2D BB of the detected objects using the frame-based detectors. The event-based masks represent the object with the most recent event in each pixel while retaining their polarity (as $+1$ for positive events and -1 for negative events). Thus, resulting in a sparse 2D matrix of integers containing integer values of 0, $+1$, and -1. Event-based masks are ideal when there is limited motion in the background of the object, causing the events generated to be mostly due to the object itself. However, it is vulnerable to events generated due to noise (global shutter or shadows) and is not ideal when the detected object (in the image) is not in motion, thus limited events are generated that can be used to generate the mask. On the other hand, edge-based masks, shown in Figure 4.3(b), rely on the actual image crop representing the object's BB. This crop is processed and converted to a binary mask representing the edges of the object, albeit without the polarity information where absolute values are used instead. Thus, the edge-based event mask is represented by a sparse 2D matrix of integers containing values of 0's and $+1$'s. The edge-based mask mimics the events generated by a moving object which are typically around its edges. Therefore, possibly making it a more appropriate choice for an object that is rapidly changing its direction. However, it is susceptible to any edges in the background and any distortions in the image due to motion blur and poor dynamic range. Overall, despite their limitations, both types of event masks can be improved by different methods, such as filtering, or generating learned masks instead of handcrafted ones. However, that is not the focus of this research but can be addressed in future works. Nevertheless, they are evaluated further in this work, in addition to the proposed methods described in the following sections.

4.3.1.4 Limitations of the framework

In summary, our prior results have proved the ability to generate high-temporal-resolution object tracking and detection by utilizing event data in combination with images. By incorporating the temporal information of the events while using a sufficient history of event data (50 ms), higher

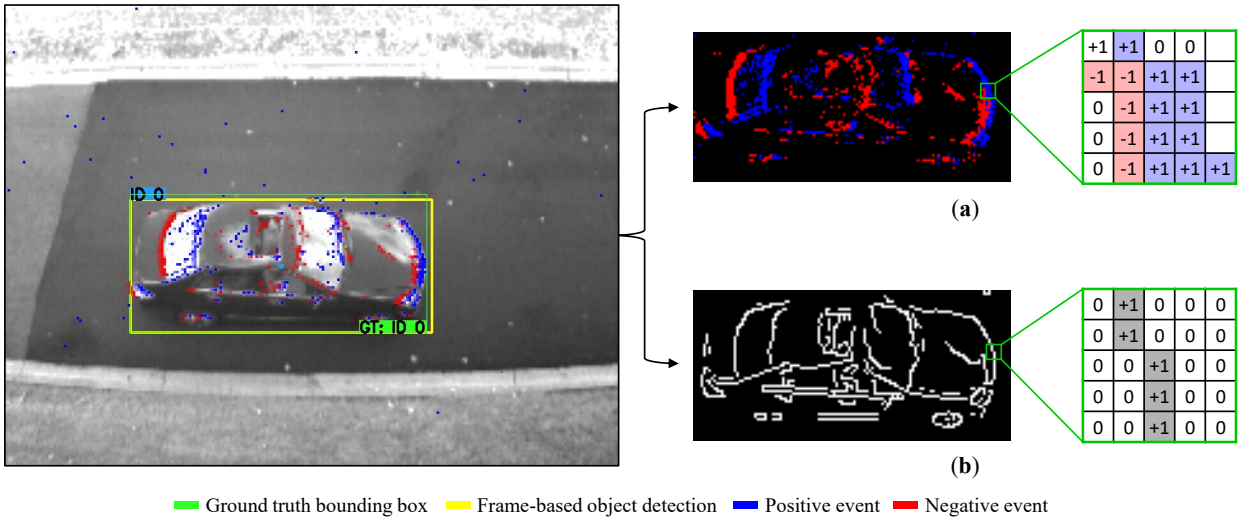


Figure 4.3: Demonstration of the two different event mask types and how they are generated after the object is initially detected in the frame, including (a) event-based mask, and (b) edge-based mask. The frame-based object detector used in this example is SSD.

temporal tracking rates (up to 384 Hz demonstrated) were feasible with very minimal performance deterioration compared to when tracking at the base framerate (24 Hz) using only image frames. By using classical computer vision techniques to amplify the tracking rates, minimal overhead is added, in comparison with the use of learned methods (such as frame-based object detectors) at higher rates.

While our general framework has shown robust results of higher temporal resolution tracking from a low sampling-rate input source with the aid of event data, it is limited by the relatively inconsistent performance of frame-based object detectors that are optimized for real-time performance (such as YOLOv3 [149]). This is exemplified in both the poor BB alignment accuracy and the commonly missed detections of objects in the scene (*i.e.*, false negatives). Poor BB estimation (also known as localization accuracy) reduces the overall detection and tracking performance, which is evident by the tracking metrics used. Likewise, missed detections cause fragments in the estimated trajectories of the tracked objects, especially as our presented framework is reliant on detecting the object in the frame domain in order to initiate the object detection process in the event domain in subsequent window frames of event data. Therefore, in this work, we present two

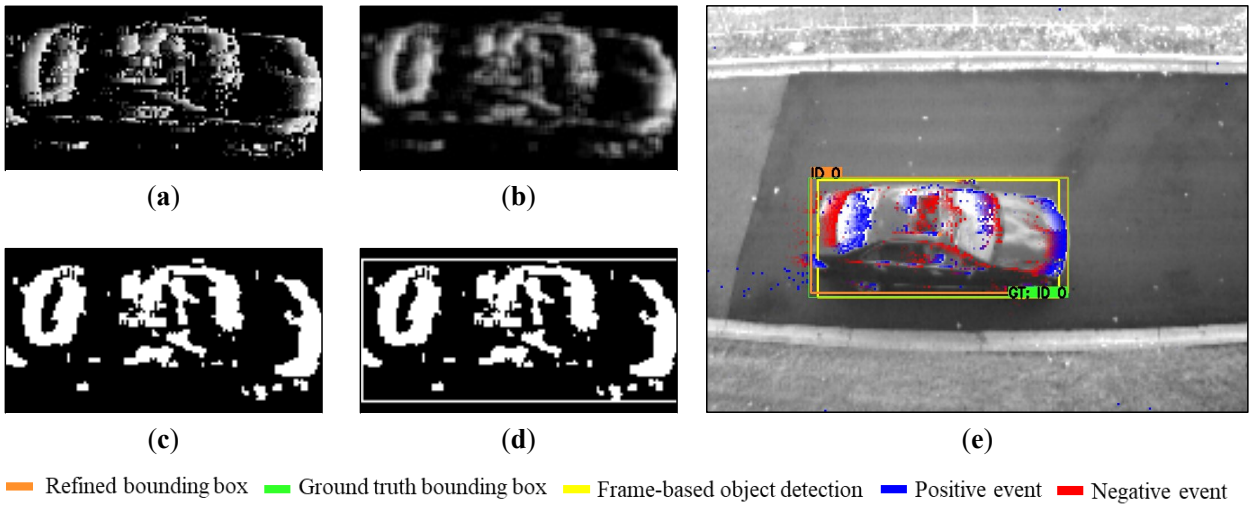


Figure 4.4: Overview of the BB refinement process of the object detected using event data. (a) Events are combined to form a grayscale image, (b) then the image is filtered to remove noise using an average blur. Finally, the resulting blurred grayscale image is then thresholded to generate a binary version, (d) where a best-fit bounding rectangle is formed that highlights the object more precisely. (e) Results are demonstrated, where the frame-based object detection and the refined BB are highlighted by the yellow and bright orange BB, respectively, while the ground truth label is shown in green.

additional methods that should improve the performance consistency of object detection (in any given frame), and the precision of the generated BB that highlights the position of the detected objects. Our prior approach along with the proposed methods are highlighted in Figure 4.2. Finally, we quantitatively verify the advantages of these enhancements with the use of state-of-the-art multi-object detection and tracking metric HOTA [111], presented in the form of an ablation study.

4.3.2 Event-based Bounding Box Refinement

Here, we explore an intuitive event-based method for BB refinement, inspired by the frame-based BB refinement methods that have been previously proposed in the literature [35]. This method is broken down into two main stages, which are event extraction and Three-dimensional (3D) matrix generation, as well as BB filtering and refinement.

4.3.2.1 Event extraction and 3D matrix generation

Once a new window frame containing an image is loaded, the frame-based object detector generates predictions, each including object classes and 2D BBs. Similar to the event mask generation process described earlier, the events available in the mask's area within the search region available in the current window frame, are extracted and added to a two-dimensional matrix. Note that the BB of the mask is a slightly enlarged version of the initial object detection's BB, thus enabling a larger initial area to freely enlarge or reduce the BB as needed based on the true size of the object. This is because frame-based object detectors might either over- or under-estimate the size of the detection's BB. Afterward, a depth channel is added to convert the resulting matrix into three dimensions (from 2D) to be able to process it like a typical image. This is followed by removing the polarity information by finding the absolute values of the events present in the matrix (negative events and positive events are treated alike).

Given that our approach incorporates temporal information, we linearly weight each event (in the 3D matrix), giving more weight (higher value) to the more recent events at that tracking time instant, similar to our approach of weighing the window frame's events temporally discussed earlier. Finally, the values of each entry in the 3D matrix are normalized to values between 0 and 1, then converted to a range of 0 and 255 representing a grayscale image as shown in Figure 4.4(a).

4.3.2.2 Bounding box filtering and refinement

Subsequently, the second stage is applied, which involves filtering the resulting grayscale image and generating a refined BB.

The resulting image from the previous stage is initially filtered by blurring it using an averaging filter with a kernel size of 3x3, as shown in Figure 4.4(b). Blurring an image is a standard method used to smoothen an image and remove noise. In our use case, we adapt this method to the grayscale image representing the events to assist in removing the ones generated due to noise, such as shadows or global shutter which is common in hybrid cameras. The resulting blurred image is then thresholded and converted to binary values (0 or 1 per each matrix's values) using Otsu's

method [131] to differentiate the object itself from the background or any noise. Otsu's method was noticed to help find a more optimal global threshold value to binarize an image appropriately which is shown in Figure 4.4(c). Finally, a best-fit BB is then generated that would precisely cover the whole object by providing the minimum contour possible to fit the object, as shown in Figure 4.4(d). The resulting BB is then fed into the object tracker to associate the detected object with ones previously tracked based on Euclidean distance [36]. Figure 4.4(e) shows the detected frame-based BB (produced by YOLOv3 [149]) against the refined and the ground truth BBs at a given time instant. We can observe that event-guided refined BB matches the ground truth more accurately than the initial frame-based detection. Note that in our prior work, the BBs generated by the frame-based object detectors are used as is without any refinement or modification, while the subsequent, inter-frame detections, would retain the same BB size as well, based on the framework described earlier in this section. Finally, this method is also applied in subsequent window frames of event data to enable BB refinement at higher temporal resolutions as well when detecting objects using the generated event masks. Thus, the BB of the detection is continuously and dynamically refined at each tracking instant.

Undoubtedly, we note that an insufficient number of events can adversely impact the outcome of this process, possibly leading to a very small, and inaccurate, BB. Therefore, it is important to set a simple check to initiate this process. Accordingly, we set a minimum threshold for the sum of the temporally weighted events, within the initial BB, to permit the refinement process. This is applied throughout our work and is reflected in our results.

Moreover, we have observed that this method can be negatively affected by more recent events generated due to noise, which are the result of some shadows or shutter noise as discussed earlier. Newer events, relative to the tracking time instant, are given more weight and thus are not always filtered out by our method. This leads these events to affect the accuracy of the best-fit BB generated in the last step. Similarly, we have noticed that more precise models, such as SSD [108], can be adversely impacted by this process, mainly due to the same type of noise events, besides the fact that the generated detections have very precise BBs, to begin with, as observed in our work.

4.3.3 Continuous Event-based Object Detection and Recovery

Efficient object detector models, such as YOLOv3, also suffer from an intermittent object detection performance. An object that appears in many consecutive frames might be detected in some but missed in others. This can affect the overall performance of the object detection and tracking given our framework's dependency on the frame-based object detector to be able to initial the inter-frame detection, as evident in our prior work's results [45]. Meanwhile, more sophisticated models, such as SSD [108], have more consistent performance, albeit at the expense of more computational requirements and greater latencies. Accordingly, to minimize the effects of intermittent frame-based object detection performance, we present a method that recovers missed detections and false negatives of objects that were previously tracked, using event data.

An event mask is generated whenever an object is detected using the frame-based object detectors. Each event mask is then used in a sliding-window mechanism to find the object's optimal inter-frame position within an enlarged area known as the search region. The highest correlating position is then assumed to be the inter-frame object position, hence the resulting event-based object detection. Nonetheless, in our prior work [45], this method was only integrated whenever an object was successfully detected in a given image to be able to perform event-based object detection before the subsequent image is read. If an object is missed, then it is no longer tracked till it is detected again by the frame-based object detector with possibly an incorrect object ID (based on the association technique). Therefore, in this work, we evaluate a similar application of using the event masks, but of previously tracked objects, to recover any missed detections at a given image frame. Thus, limiting the gaps in the estimated objects' trajectories and improving the overall object detection and tracking performance.

We describe this process as follows. Given a set of previously tracked objects, an undetected object in the image, within the latest window frame, is marked as "disappeared" after associating all of the frame's detections using the object tracker. The missed object's latest event mask, utilized in the previous window frame, is similarly used to initiate the event-based object detection process using the events available in the latest window frame, located within the search region. This

follows the same process utilized in our prior work for inter-frame event-based object detection. Nevertheless, due to lower certainty, a significantly higher event-based detection score threshold is used. This threshold resembles the minimum matrix multiplication summation value that must be met or surpassed in order to consider the object detected in the event domain. We apply this validation process to limit the detections that may result from events produced by other objects or generated by noise, given the lower detection confidence to begin with, and considering that this method is neither dynamic nor learned, unlike the frame-based object detectors we use in our framework.

As with the former method, this one has its limitations and drawbacks as well, if not properly optimized. This method can be negatively affected by several factors, such as false positives (incorrect detections) or poorly aligned detections generated by frame-based object detectors, as shown in Figure 4.5(a), which are given higher precedence and are prioritized over the event-based methods, given their presumed robust and dynamic performance. These false positives, even if intermittent or sporadic, would be used for continuous event-based detection, as described earlier, to recover the missed detections. Based on the event mask generation process, these false detections can lead to consistent and continuous false detections in subsequent window frames if subjected to a sufficient number of events generated due to noise or other external factors, as shown in Figure 4.5. Thus, affecting the overall detection and tracking accuracy.

4.3.4 Ablation Study

To verify the feasibility and the benefits of the proposed methods, we conduct an ablation study to assess the influence of each on the overall object detection and tracking operation and to indicate the right combination of methods to achieve optimal performance. Moreover, we provide a computational latency analysis of the estimated average latency of each core component of our framework.

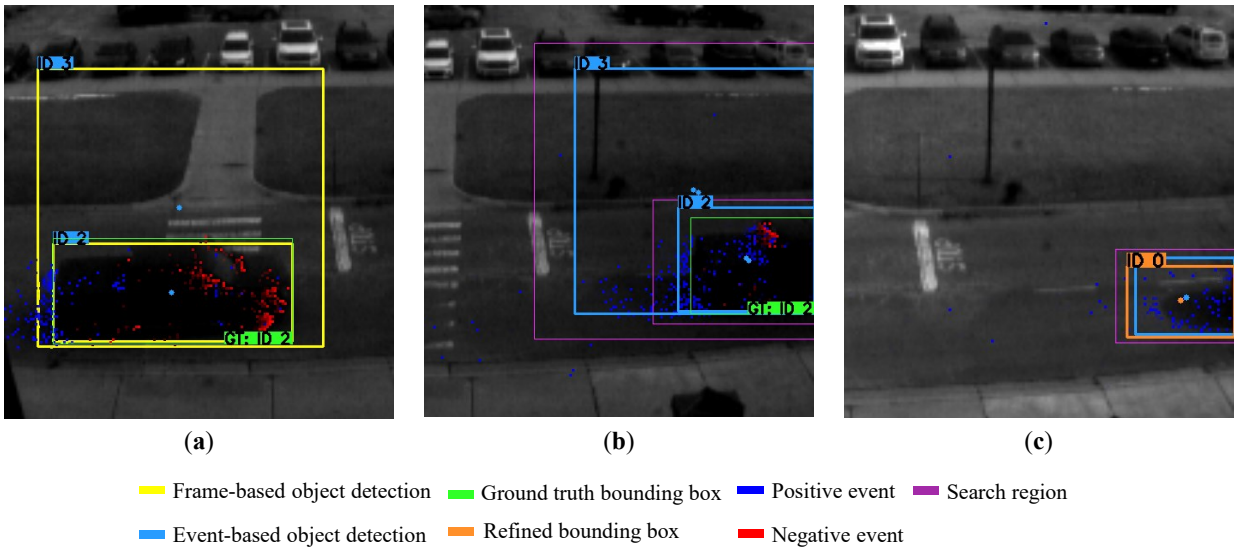


Figure 4.5: Demonstration of some failure modes when accompanied by continuous event-based object detection and BB refinement: (a) false object detection generated by the frame-based object detector SSD, (b) which are then continuously yet incorrectly detected using events in subsequent window frames; (c) an object is falsely detected and recovered due to events generated by the shadow of a vehicle leaving the scene.

4.3.4.1 Evaluation parameters and configurations used

Similar to the setup used in our previous work [45], we use two pre-trained frame-based object detectors which are YOLOv3 [149] and SSD [108]. We select the YOLOv3-320 variant trained on Microsoft COCO labels [103], with an input image size of $320 \times 320 \times 3$; and the VGG16-based SSD-300 variant, trained on PASCAL VOC labels [50], with an input image size of $300 \times 300 \times 3$. The models selected propose a trade-off between latency and accuracy. Moreover, we set both confidence and non-maximal suppression thresholds are set to 50%. Only predictions related to the 'vehicle' class (and its different forms) are considered, whereas other object classes are simply ignored and filtered out. Both frame-based object detector models are used along with different detection and tracking configurations described in this section.

Furthermore, we build on the two best approaches presented earlier [45] with the proposed methods presented earlier, to verify the validity of our assumptions. The two best approaches incorporated a moving window frame of image and event data, of the last 50 ms which are tem-

porally weighted, at any given tracking instant, while only varying by the use of different event mask types (event-based and edge-based described in Section 4.3) for event-based object detection. These modes are referred to as modes 2 and 3, for event-based and edge-based masks, respectively. The combinations relating to each of these modes are referred to as A(1–3) for mode 2, and B(1–3) for mode 3. Moreover, we provide the results of the single-modal, frame-based approach for object detection and tracking (without any event-based methods), to provide a baseline reference to compare with the proposed improvements at the preset rate of 24 Hz.

In this ablation study, the evaluation is conducted using the MEVDT dataset, as detailed in Chapter 2. This dataset, captured using the DAVIS 240c sensor, comprises 63 sequences with fully labeled vehicle objects, including BBs and object IDs, across various tracking rates (24, 48, 96, 192, and 384 Hz) as described in Chapter 3. This range of rates allows for the quantitative assessment of performance across different temporal resolutions. The DAVIS 240c, featuring both an APS and a DVS using the same pixel array, captures grayscale images at 24 Hz and events with a 1 mus resolution. The data was collected from a static camera setup, simulating an infrastructure camera setting, capturing vehicles in motion at varying speeds and accelerations. This dataset is instrumental in evaluating the performance of our object detection and tracking configurations, as it provides several scenarios and tracking rates for robust testing.

Finally, the well-defined object detection and tracking metric HOTA [111] is used to evaluate the overall object detection and tracking performance of the different tracking configurations. Compared to prior metrics in the literature, the HOTA metric provides a good balance between the overall object detection, association, and localization accuracy in a combined metric. In addition, HOTA can be decomposed into a series of sub-metrics that describe each. These sub-metrics include the Detection Accuracy (DetA), which describes how well detections are aligned; the Association Accuracy (AccA) which measures how well matched-object trajectories are aligned and associated across time; and the Localization Accuracy (LocA), which refers to how well spatial alignment is between the predicted and ground truth detection. The main metric HOTA, as well as the sub-metrics, are calculated over a range of intersection-over-union threshold α values, span-

Table 4.1: Results of the ablation study on the influence of each proposed feature at different combinations while using YOLOv3[149] as the frame-based object detector. HOTA[111] metrics are used to numerically assess the performance of these combinations at the base framerate of 24 Hz and an elevated tracking rate of 384 Hz with significantly higher temporal resolution. At both tracking rates, results show optimal performance was achieved when combining both proposed features.

Object Detector	Tracking Rate	Tracking Mode	Proposed features		Mask type		Metrics (%)						
			Bounding Box Refinement	Continuous Event-based Detection	Event-Based	Edge-Based	HOTA	DetA	AssA	LocA	HOTA(0)	LocA(0)	HOTA-LocA(0)
YOLOv3	24 Hz	*†	-	-	-	-	56.6	53.0	60.8	84.2	68.1	82.0	55.9
		A1	✓		✓		59.3	56.1	62.9	**87.9**	68.1	**86.4**	58.9
		A2		✓	✓		60.1	60.1	60.6	83.1	77.1	78.8	60.8
		A3	✓	✓	✓		**64.1**	**64.5**	**63.8**	86.4	**77.4**	83.5	**64.6**
		B1	✓			✓	59.3	56.1	62.9	**87.9**	68.1	**86.4**	58.9
		B2		✓		✓	59.9	58.2	62.1	83.4	75.4	79.6	60.0
		B3	✓	✓		✓	**64.9**	**64.1**	**65.9**	86.8	**77.6**	84.3	**65.4**
	384 Hz	2†			✓		52.5	50.1	55.3	84.1	63.8	81.3	51.9
		A1	✓		✓		54.6	52.3	57.1	**86.9**	63.9	**84.5**	54.0
		A2		✓	✓		59.3	56.0	63.1	83.1	**75.8**	78.6	59.6
		A3	✓	✓	✓		**63.1**	**59.7**	**66.8**	86.1	**76.0**	83.1	**63.2**
		3†				✓	51.3	48.9	54.1	84.2	62.3	81.5	50.8
		B1	✓			✓	53.1	50.6	55.8	**86.6**	62.3	**84.2**	52.5
		B2		✓		✓	54.6	51.3	58.4	83.4	68.8	79.2	54.5
		B3	✓	✓		✓	**60.2**	**57.3**	**63.5**	85.9	72.4	83.1	**60.2**

* Single-modal image-only tracking (excludes event data).† Results from Table 3.1. The top two results per tracking rate and metric are highlighted in **bold**.

ning from 0.05 to 0.95 with increments of 0.05. The HOTA metrics are described in further detail in Ref. [111]. Further, we note that the output was recorded and saved in the MOTChallenge format [38] where it is used to calculate the final detection and tracking metrics using TrackEval [82], developed by J. Luiten. For compactness, we report the results of only two tracking rates, 24 and 384 Hz, given that they should provide both ends of the performance spectrum, where intermediate tracking rates are expected to perform within that range.

4.3.4.2 Ablation study results

The results of the ablation study are presented in Table 4.1 and Table 4.2 for the frame-based object detectors YOLOv3 [149] and SSD [108], respectively.

In Table 4.1, results show significant performance advantages when both methods are utilized with either event-mask type. At 24 Hz, we notice that the DetA improves from 53% to 64.5% and

Table 4.2: Results of the ablation study on the influence of each proposed feature at different combinations while using SSD [108] as the frame-based object detector. HOTA [111] metrics are used to numerically assess the performance of these combinations at the base framerate of 24 Hz and an elevated tracking rate of 384 Hz with significantly higher temporal resolution. At both tracking rates, results show optimal performance was achieved when combining both proposed features.

Object Detector	Tracking Rate	Tracking Mode	Proposed features		Mask type		Metrics (%)						
			Bounding Box Refinement	Continuous Event-based Detection	Event-Based	Edge-Based	HOTA	DetA	AssA	LocA	HOTA(0)	LocA(0)	HOTA-LocA(0)
SSD	24 Hz	*†	-	-	-	-	69.0	67.4	70.9	**89.1**	77.2	87.9	67.9
		A1	✓		✓		67.3	66.6	68.1	88.1	76.8	**86.7**	66.6
		A2		✓	✓		66.8	63.9	**70.2**	88.0	78.7	85.0	66.9
		A3	✓	✓	✓		**69.0**	**68.6**	69.6	87.0	**82.5**	84.3	**69.5**
		B1	✓			✓	67.3	66.6	68.1	88.1	76.8	**86.7**	66.6
		B2		✓		✓	**69.2**	67.7	**71.0**	88.3	79.9	86.0	**68.7**
		B3	✓	✓		✓	68.5	**68.1**	69.0	87.4	**80.2**	85.3	68.4
	384 Hz	2†			✓		65.0	62.5	67.8	**88.8**	73.2	**87.0**	63.7
		A1	✓		✓		63.0	61.1	65.1	87.3	73.0	85.1	62.1
		A2		✓	✓		**65.9**	62.0	**70.2**	87.9	**77.6**	84.6	**65.6**
		A3	✓	✓	✓		**66.4**	**63.9**	**69.2**	86.5	**79.4**	83.6	**66.4**
		3†				✓	62.5	60.4	64.7	**88.7**	70.6	**86.9**	61.3
		B1	✓			✓	60.4	58.9	62.1	86.9	70.3	84.7	59.6
		B2		✓		✓	63.8	60.2	67.7	88.0	74.1	85.4	63.2
		B3	✓	✓		✓	63.3	60.7	66.1	86.3	75.3	83.6	63.0

* Single-modal image-only tracking (excludes event data).† Results from Table 3.2. The top two results per tracking rate and metric are highlighted in **bold**.

64.1% for the event-based (A3) and edge-based (B3) masks, respectively. Similarly, the AssA improves from 60.8% to 63.8% and 65.9% under the event mask types as well, with the final HOTA equal to 64.1% and 64.9%. Further, we observe that the LocA performance of YOLOv3 has significantly improved with the introduction of the BB refinement methods, as indicated by A1 and B1, and the combined methods A3 and B3. Meanwhile, at the higher tracking rate of 384 Hz, the combination of the two features presents the most performance gain consistent with the results at 24 Hz, showing significant performance gains over our previous work's results (as indicated by modes 2 and 3) under all the metrics. This is highlighted by the substantial improvement in the general HOTA metric values from 52.5% to 63.1% and from 51.3% to 60.2%, when incorporating both methods under the event-based and edge-based masks, respectively. Thus, showing significant performance advantages for optimized object detectors and even more limited performance deterioration at higher temporal resolutions for object tracking. Note that the performance vari-

ance at 384 Hz can be attributed to the inconsistent frame-based object detection performance of YOLOv3, which results in frequent detection gaps in the images. As a result, based on our framework's design, these frame-based detections are required to initiate the inter-frame event-based detection process. Therefore, such gaps can be substantially amplified at higher tracking rates. For example, there would be one window frame that contains an image for every 15 window frames that contain only events, at a tracking rate of 384 Hz with a 24 Hz camera framerate. Accordingly, a missed detection in a given image can cause a missed detection, and gaps in the object tracking trajectory, for a total of 16 consecutive window frames. This further emphasizes the importance of the continuous event-based object detection process introduced in this work.

Conversely, our proposed methods had slightly less performance improvement when using SSD as the frame-based object detector, as shown in Table 4.2. At 24 Hz, we show that the DetA marginally improves from 67.4% to 68.6% and 68.1% for the combined features using an event-based and edge-based mask, respectively, as indicated by modes A3 and B3, whereas the AssA was already high, to begin with, and did not show any meaningful benefit. This is due to the relative simplicity of the dataset used (few occlusions and a low number of objects simultaneously at any time instant), as well as the robust object detection performance of the SSD variant used in this study. Furthermore, the LocA was a bit adversely impacted by either combination of the proposed methods. This demonstrates the limitations of the BB refinement method which can negatively affect the BB precision of a robust object detector such as SSD, as discussed in Section 4.3. Nevertheless, the more comprehensive metric, HOTA, showed consistent performance in A3, and slightly improved performance using the edge-based mask combined with the continuous event-based detection feature. On the other hand, at 384 Hz, the proposed methods showed respectable improvements to the prior results [45]. Both the DetA and the AssA, as well as the general HOTA values, showed improvements, with the A3 configuration yielding the best performance at the tracking rate of 384 Hz. Overall, for SSD, the proposed methods had mixed effects initially but were more beneficial at higher tracking rates.

Finally, we provide some qualitative results of our system at the baseline tracking rate of 24 Hz,

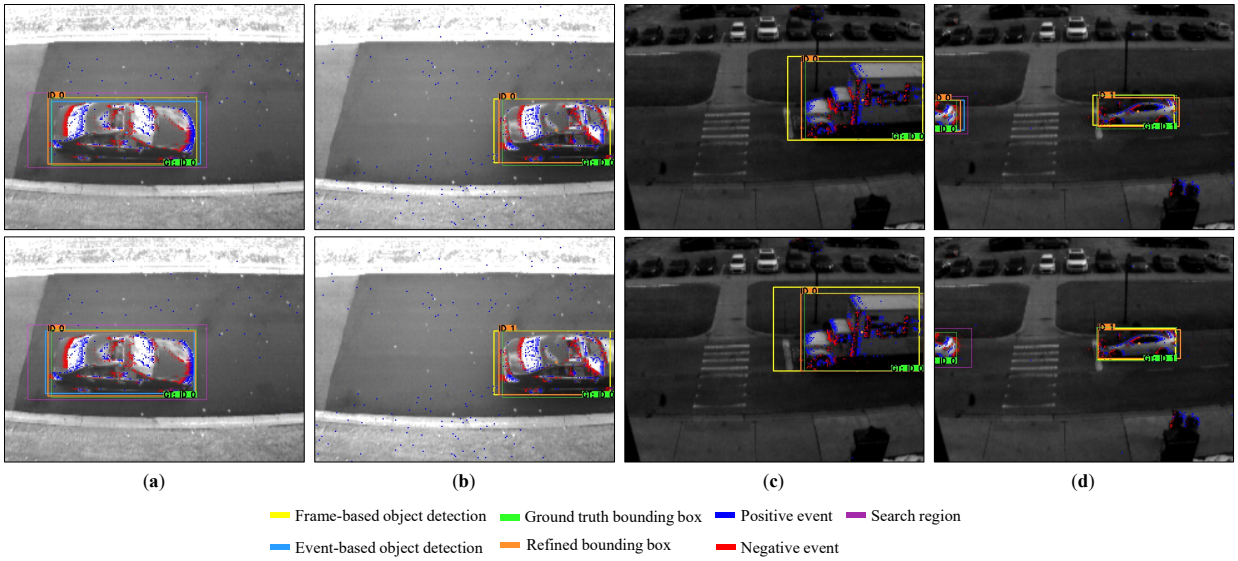


Figure 4.6: Qualitative results of our multi-modal object detection and tracking framework at the baseline tracking rate of 24 Hz at various instances. The top and bottom rows represent the same time instant but for different configurations. (a) and (b) demonstrate the output of YOLOv3 with A3 (top) and B3 (bottom) event-based configurations, whereas (c) and (d) compare the A3 configuration for both YOLOv3 (top) and SSD (bottom). This demonstrates the effects of the proposed methods in combination with our multi-modal framework.

as shown in Figure 4.6, to visually demonstrate the effects of the proposed methods under different configurations and settings. For example, Figure 4.6(a) shows an instance where the frame-based object detector used, YOLOv3, missed detecting the vehicle in the scene. Instead, this object, which was previously tracked, is detected using the event-based method for configurations A3 (top) and B3 (bottom), whereas (b) shows the BB refinement process which generates a more accurate BB, relative to the ground truth label, compared to the initial frame-based object detection for the same configuration. In the same manner, Figure 4.6(c) and (d) show the output of the A3 tracking configuration for both YOLOv3 (top) and SSD (bottom), showing multiple tracked objects with the application of the BB refinement process, in addition to the event-based object detection method that recovered an object in the YOLOv3 configuration (top) but failed in the SSD configuration (bottom) as demonstrated in Figure 4.6 (d).

4.3.4.3 Computational latency analysis

To analyze the overall performance and the computational requirements of the system, we provide a simple computational latency analysis for the different core components of the system on a relative basis, as demonstrated in Table 4.3. Note that all the tests, in our analysis, were implemented and conducted on a CPU, specifically an Intel i7-7700HQ. Therefore, a GPU was not used in our testing. Moreover, the event-based methods, at this stage, are not optimized for runtime performance (*i.e.*, no multi-threading or compiled libraries) given that this work was implemented using the scripting language, Python. We also note that, due to their classical design, the computational latency of the presented event-based methods is linearly proportional to the object's size. Accordingly, we present the results per one object for the detection and tracking process, where the average BB size was found to be around 80×45 in these tests. Consequently, this analysis is meant to highlight the difference in the computational requirements and average latencies of the event-based and the frame-based components as well in a relative manner. This leaves significant room for future work in terms of optimizing this framework for better real-time performance, which is not the focus of this work.

Overall, the results in Table 4.3 show a significant disparity between the latency of the frame-based object detection components and the event-based components. This can be attributed to the scale and complexity of the learned components used. As for the difference in the average latency of both frame-based object detectors, we note that YOLOv3's [149] architecture utilizes the DarkNet-53 feature encoder as its backbone which has 42 million parameters [37]. Meanwhile, SSD-300's architecture uses VGG-19 [160] instead, which has over 143 million parameters, thus, justifying the 134 ms variance in their average latency, and the difference in their single-modal object detection and tracking performance (at 24 Hz) as demonstrated in Table 4.1 and Table 4.2. In contrast, event-based methods presented have shown at least an order of magnitude less latency than their frame-based counterpart. For instance, the initial event mask generation process takes only 0.66 and 0.26 milliseconds, on average, for the event-based and edge-based masks, respectively. This is followed by the inter-frame, event-based, object detection process, which

Table 4.3: Computational latency analysis of the main stages of the proposed multi-modal object detection and tracking framework on a relative basis, using only a CPU. Event-based methods demonstrate at least an order of magnitude lower latency than the frame-based object detectors' inference times. The estimated total latency refers to the worst-case scenario, where a window frame contains an image of an object that was initially missed, then detected using the event-based method, based on the specified configuration, while refining the resulting BB. The event-based methods are estimated per single object with an average BB size of 80×45 pixels.

Stage	**Method**	**Average Latency (ms)**
Frame-based Object Detection	SSD-300 (VGG16) [108]	359
	YOLOv3-320 (DarkNet53) [149]	225
Event Mask Generation	Event-based mask	0.66
	Edge-based mask	0.27
Event-based Object Detection	using Event-based mask	23.7
	using Edge-based mask	29.0
Additional Methods	Bounding box refinement	0.37
	Continuous detection*	23.7–29.0
Total Latency	A3 SSD	384
	B3 SSD	389
	A3 YOLOv3	250
	B3 YOLOv3	256

* Equivalent to the event-based object detection's latency depending on the type of the event mask used.

takes 23.7 ms, on average, when using the event-based mask, and 29 ms when using the edge-based mask. Therefore, the combined, multi-modal approach presented in this chapter can enable high-temporal-resolution object detection and tracking results with minimal overhead using a low-framerate camera, compared to the single-modal approach that incorporates a high-framerate camera with DNN-based frame-based object detectors.

As for the computational overhead resulting from the additional methods presented, we notice that the BB refinement process provides a very low-cost solution that delivers noticeable performance improvements with just 0.37 ms of average latency. Meanwhile, the continuous event-based object detection process has an overhead equal to the main inter-frame object detection process, given that they utilize the same procedure. However, this method is only used when a previously

tracked object was not detected in a given image, thus resorting to the asynchronous event domain in an attempt to recover it. Accordingly, we estimate the worst-case total latency possible for our multi-modal framework at the baseline camera framerate of 24 Hz. This scenario assumes that an object was not initially detected by the frame-based object detector selected, followed by the process of event-based object detection, in addition to the BB refinement process, as represented by the A3 and B3 tracking modes described earlier in this section. The results show that the base performance of YOLOv3, in addition to the event-based methods presented, with minimal computational overhead based on our framework and a total latency of around 250 ms, can compete with the performance of the single-modal, frame-based, tracking-by-detection approach that uses SSD with 359 ms latency ($>$100 ms difference).

In summary, the results presented validate our assumptions, especially under more efficient and optimized DNN object detectors such as YOLOv3. Our results showed significant overall object detection and tracking improvements using classical computer vision techniques that leverage event data. The best performance was achieved when combining both proposed methods under either event mask type, especially when using the event-based mask. The outperformance of the event-based mask further proves the benefit of incorporating the polarity data of events in object detection and tracking applications.

4.4 Experiment Design

Building on the MEVDT dataset introduced in Chapter 2, we design an experiment to evaluate vehicle detection and tracking performance using high-temporal-resolution LiDAR measurements. This experiment aims to validate the effectiveness of our event-based approaches in real-world vehicle position-tracking scenarios. This section describes the design of this experiment, including how the data is collected and processed, as well as the metrics used in our evaluation.

4.4.1 LiDAR-based Tracking Experiment Setup

For this experiment, we employ the industrial high-speed LiDAR Benewake TF03-100, as briefly mentioned in Chapter 2. This LiDAR is capable of measurements up to 100 meters with a resolution of 1 cm and an update rate of 1000 Hz and has an estimated error of $\pm$10 cm within the 10 m range, and about 1% error beyond 10 m. It was used to capture high-temporal-resolution ground truth positional measurements of vehicles in 13 out of the 63 recorded dataset sequences. Specifically, it was set up to track a single vehicle driving towards it at varying speeds, positioned at a range of 30 to 60 meters. Concurrently, the DAVIS 240 camera, positioned on a high elevation and pointing downwards, captured both the vehicle's images and events. This setup allowed for synchronized data collection from both the LiDAR and the camera, enabling a precise analysis of vehicle movement. Based on our measurements, the vehicle enters the camera's field of view (in Scene B) approximately at the 40-meter mark and exits at around 28 meters from the LiDAR.

4.4.2 Data preprocessing and Synchronization

To make use of the data collected to estimate the performance of our tracking methods, the data from the different modalities must be preprocessed and synchronized to yield useful data.

4.4.2.1 Ground truth LiDAR data processing

After recording the distance measurements using LiDAR, the gaps due to the sporadic missing data points of the undetected vehicle are initially filled using linear interpolation. Afterward, the data points are resampled to convert from a non-uniform to a uniform sampling rate of 1000 Hz, equivalent to a 1 ms difference between every two consecutive data points. Finally, the resulting data is smoothened and filtered using a Chebyshev Type II-type low-pass filter with a passband and a stopband frequency of 10 and 12 Hz, respectively, as well as a passband ripple of 1dB and a stopband attenuation of 80 dB. Thus, generating pre-processed ground truth distance measurements with a sampling rate of 1000 Hz. The resulting synchronized and filtered LiDAR-based ground

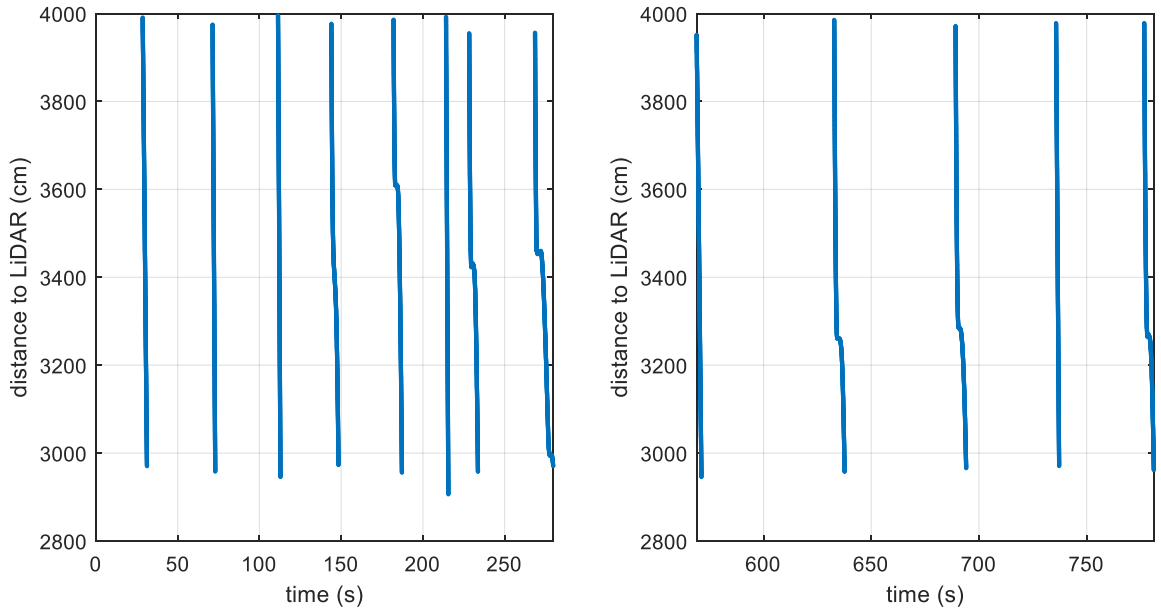


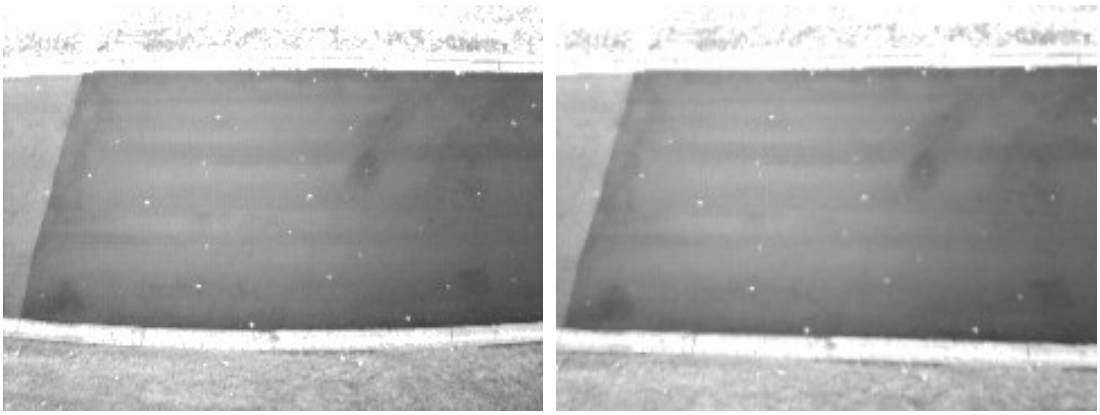
Figure 4.7: Filtered LiDAR ground truth distance data collected with a sampling rate of 1000 Hz showing 13 trajectories for the single-object tracking validation experiment. The time between 300 and 500 seconds is removed due to no available tracking data.

truth data is demonstrated in Figure 4.7.

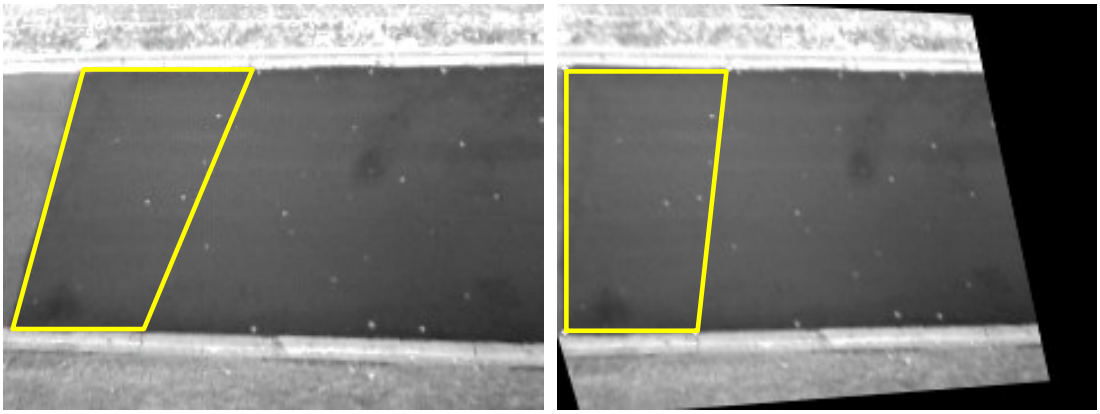
4.4.2.2 Converting 2D Object tracking to distance measurements

To convert the 2D BBs of the vehicle detections into estimated distance measurements, we use the center-right coordinate $V_{(x,y)}$ of the detection's BB at a given moment, resembling the front of the vehicle. This coordinate can then be used to estimate the distance from the LiDAR. However, due to the different types of lens distortion, namely radial and tangential, such conversion is non-linear. Hence, the removal of lens distortion is critical for accurate positional estimation. This is achieved by the camera calibration process which generates the camera's geometry (also known as intrinsic parameters) as well as the lens distortion models which are used to correct the captured images, as the event coordinates, as shown in Figure 4.8(a).

Afterward, to enable a linear pixel position-to-distance conversion, the bird's eye-view perspective is a common approach to object position estimation, especially in traffic surveillance applica-



(**a**) Lens distortion removal.



(**b**) Perspective transform.

Figure 4.8: Demonstration of the camera calibration process and bird's eye perspective transform which are required for accurate object tracking and positional measurements. (**a**) the distortion is removed from the captured image using the DAVIS 240c sensor, (**b**) then the scene's perspective is transformed to provide a bird's eye view of the region of interest.

tions. However, due to the geometry of the road relative to the camera and its elevation level, a perspective transform is necessary to yield a linear conversion. Accordingly, the perspective transformation matrix M of size 3×3 is generated using 4 initial points (pixels) on the image, along with the desired final coordinates for each of them, to produce a bird's eye view perspective, as shown in Figure 4.8(b). Moreover, the 4 chosen points are of known real-world distance measurements, which enable the final conversion from a pixel coordinate, on the x-axis, to a distance-per-pixel value.

We note that this process is performed initially to yield the required transformation matrices

where the tracking point $V_{(x,y)}$, which refers to the center-right coordinate of the detected vehicle, is initially undistorted, then transformed using the following equation:

$$Z_{(x,y)} = \left(\frac{M_{(0,0)}V_x + M_{(0,1)}V_y + M_{(0,2)}}{M_{(2,0)}V_x + M_{(2,1)}V_y + M_{(2,2)}}, \frac{M_{(1,0)}V_x + M_{(1,1)}V_y + M_{(1,2)}}{M_{(2,0)}V_x + M_{(2,1)}V_y + M_{(2,2)}} \right), \tag{4.1}$$

where V_x and V_y and the undistorted x and y coordinates of the tracking point $V_{(x,y)}$, M is the perspective transform matrix generated earlier, and $Z_{(x,y)}$ is the final transformed pixel position. Note that the center-right coordinate, $V_{(x,y)}$, resembles the front of the vehicle. Accordingly, the x-coordinate of the resulting point $Z_{(x,y)}$ is converted to the estimated distance to the LiDAR by normalizing it (subtracting from and dividing by the frame's width in pixel count) and then multiplying by the measured distance-per-pixel ($\sim$4.4 cm/pixel). Finally, the resulting point is offset by the estimated distance between the frame's side to the LiDAR (28 m). This process is applied to all the tracking points produced by our methods presented in Section 4.3.

4.4.2.3 Temporal synchronization

The system clocks of the different computers are typically synchronized using an online global clock, however, only to the order of seconds at best. Moreover, in our data collection procedure, a convenient network-based time synchronization method was not feasible due to the significant distance between both the camera and the LiDAR, eliminating the possibility of conveniently connecting the data collection computers to enable temporal synchronization. This presents a challenge for our high-temporal-resolution tracking data when evaluating it according to the ground truth distance. Therefore, we intuitively achieve temporal synchronization by using the previously labeled 2D ground truth BBs, generated at a high rate of 384 Hz [45], for the 13 sequences involved, and manually synchronizing them with a resampled version of the filtered LiDAR data collected at 1000 Hz presented earlier, thus minimizing the temporal difference as much as possible. Accordingly, we have found the temporal synchronization difference to be 0.719 seconds between the system clocks of both devices. This value is subsequently used to offset all of the generated vehicle tracking data and enable a millisecond level of synchronization.

4.4.3 Experiment Parameters and Metrics

4.4.3.1 Vehicle detection and tracking configurations

Using the optimal configuration, found in our ablation study in Section 4.3, which combines both improvements proposed, we evaluate the distance estimation results at different uniform rates of 24, 50, 100, 200, and 500 Hz, using the tracking modes A3 and B3 described earlier, in combination with the frame-based object detectors YOLOv3 [149] and SSD [108]. We match the ground truth LiDAR data to the different tracking rates by resampling it at the defined uniform rates, while only keeping the tracking data where the vehicle is in the camera's field of view till just before it starts leaving the scene (*i.e.*, the front of the vehicle no longer visible in the frame). Then, the resulting outliers are manually removed, if any, which are due to the interpolation required in the resampling process of ground truth data, to yield an identical number of possible tracking points at each tracking rate for proper and accurate evaluation.

4.4.3.2 Evaluation metrics

The results of this experiment are evaluated using several error metrics, including the median absolute error, the median relative tracking error, and the root-mean-square error (RMSE). The metrics are estimated using only the successfully detected points, which are highlighted by the successful detection rates calculated for each tracking configuration. The median error metrics are chosen due to their robustness to outliers in comparison to the mean error metrics. Such outliers can result due to the resampling and temporal synchronization errors, which affect the results but without validity. Further, we note that this experiment does not account for any additional object detections besides the vehicle being tracked. These detections are ignored. However, missed detections (false negatives) would affect the vehicle detection success rates. Finally, we provide the mean of the temporal synchronization errors at each rate to provide insight into their effects on tracking performance, if such correlation occurs.

4.5 Results and Discussion

The results of the validation experiment are presented in Table 4.4. To begin with, we can notice that our presented event-based methods, in combination with our general framework, are advantageous to the vehicle detection and tracking process as highlighted in the lowest tracking rate. In terms of successful detection rates, at 24 Hz (equivalent to the image frames sampling rate), the vehicle detection rate improved significantly from 66.4% to 81.3% when using our event-based object detection recovery presented earlier, compared to only frame-based detection when using YOLOv3, further validating our prior assumptions. Similarly, detection results improved, albeit marginally, by 0.1% when using SSD as the frame-based object detector. This is due to the selected SSD variant being a more accurate and stable object detector. Meanwhile, at higher tracking rates, the event-based mask approach A3 has increasingly outperformed the edge-based mask approach B3, ending with a difference of 6.2% for YOLOv3 and 7.1% for SSD at 500 Hz, compared to a negligible difference at 24 Hz.

As for the main error metrics, we observe that modes A3 and B3 have presented significant benefits when combined with YOLOv3 at 24 Hz, but with mixed results for SSD at the same rate. Nevertheless, at higher tracking rates, the error magnitudes improve substantially for either object detector, where the best multi-modal tracking mode, A3, at 500 Hz resulted in a median absolute error of 6.4 cm and 4.8 cm, for YOLOv3 and SSD, respectively.

Interestingly, we observe that the error rates get consistently lower as we increase the tracking rates, with a negligible deterioration in object detection performance, when using either frame-based object detectors with any event-based object detection mode. This can be partially attributed to the BB refinement process which improves the accuracy of the point $V_{(x,y)}$ that represents the front of the vehicle, where if given more tracking points, it would consistently improve on each. Another reason could be the higher average temporal synchronization error, indicated at the lower tracking rates, which is primarily due to the resampling and synchronization errors of the ground truth data collected at 1000 Hz when downsampled to lower tracking rates.

Moreover, in Figure 4.9, we plot the results for 4 selected trajectories (sequences 3, 12, 7, and

Table 4.4: Vehicle detection and tracking validation experiment results under various detection configurations and tracking rates. A summary of different error metrics is presented, along with the successful detection rates and temporal synchronization errors. We demonstrate that our presented methods successfully leverage event data to enhance vehicle detection and tracking performance at various tracking rates.

Frame-based Object Detector	Tracking Rate (Hz)	Event-based Detection Mode	Metric				
			Median Abs. Error (cm)	Median Relative Error (%)	RMSE (cm)	Successful Detection Rate (%)	Mean Temporal Synchronization Error (s)
YOLOv3	**24**	N/A*	21.2	0.66	31.0	66.3	0.0112
		A3	11.1	0.33	27.8	**81.2**	
		B3	11.4	0.34	22.9	79.6	
	50	A3	9.2	0.28	17.9	80.8	0.0043
		B3	10.4	0.31	19.0	76.2	
	100	A3	7.2	0.22	15.4	81.1	0.0016
		B3	8.9	0.27	17.1	75.4	
	200	A3	6.7	0.20	14.3	81.0	0.0013
		B3	8.4	0.25	16.3	75.2	
	500	A3	**6.4**	**0.19**	**13.6**	81.1	**0.0004**
		B3	8.1	0.25	15.9	74.9	
SSD	**24**	N/A*	6.6	0.20	59.9	91.2	0.0112
		A3	7.2	0.22	21.7	**91.3**	
		B3	7.2	0.22	21.7	**91.3**	
	50	A3	6.2	0.18	14.8	90.5	0.0043
		B3	6.4	0.20	14.8	85.9	
	100	A3	5.2	0.16	11.4	90.3	0.0016
		B3	5.5	0.17	12.7	84.3	
	200	A3	5.0	**0.15**	11.2	90.5	0.0013
		B3	5.2	0.16	11.9	83.7	
	500	A3	**4.8**	**0.15**	**10.0**	90.4	**0.0004**
		B3	5.0	**0.15**	11.5	83.3	

* Object detection and tracking using image frames only (no event data used). The best result per detector and metric is highlighted in **bold**.

11) out of the 13 presented in Figure 4.7. The results of the modes A3 and B3, at the tracking rates of 24 Hz and 500 Hz, are compared to the ground truth data and the frame-based-only tracking results at 24 Hz, using YOLOv3 as the frame-based object detector. The selected trajectories represent different vehicle acceleration rates, where the vehicle moves with a consistent speed in trajectories (a) and (b) in Figure 4.9, whereas it comes to a full stop before accelerating again in trajectories (c) and (d). Similarly, the results demonstrate the feasibility of high-temporal-resolution tracking, as well as the improvement to the frame-based method at 24 Hz. Nonetheless, we notice

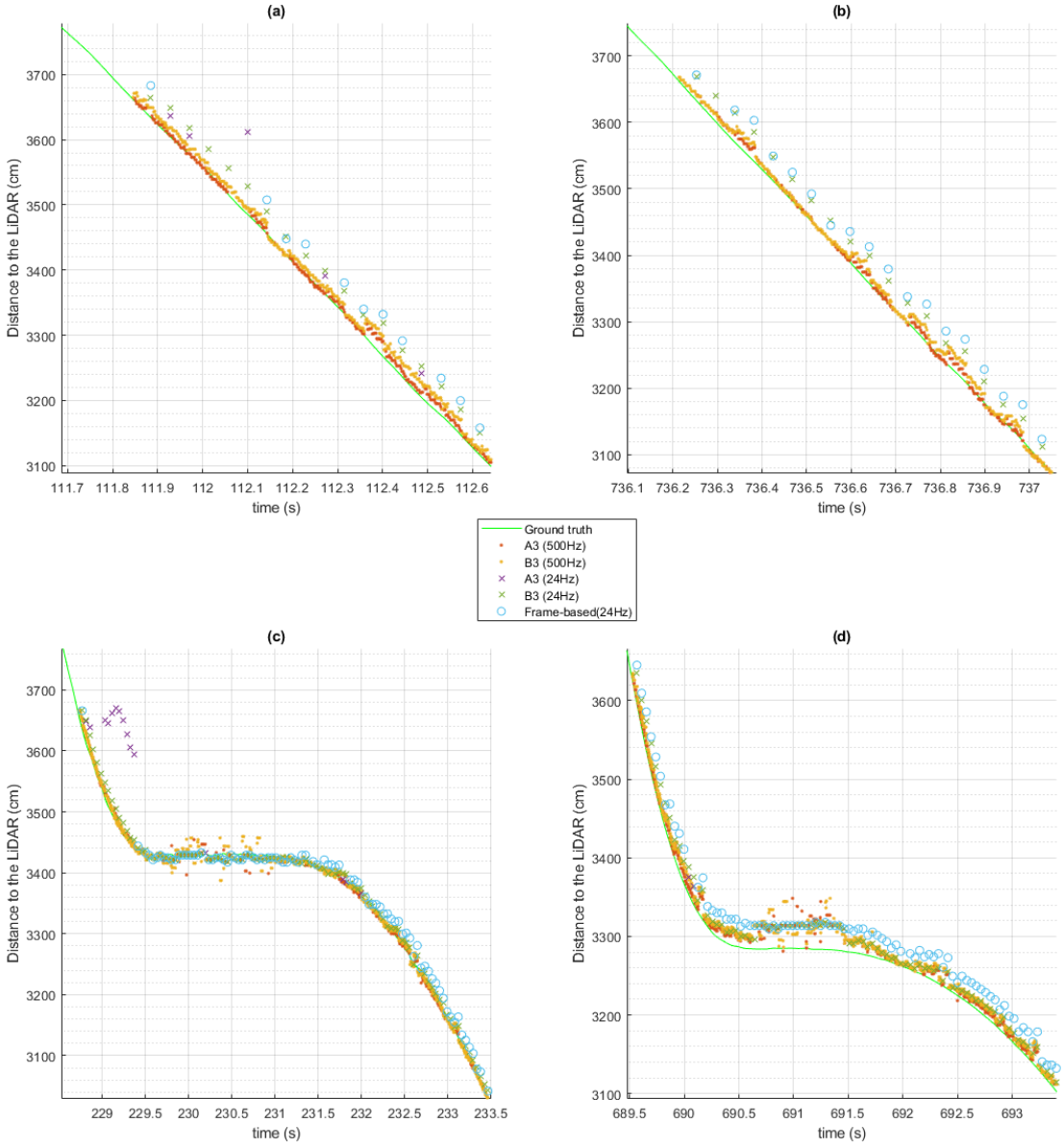


Figure 4.9: Demonstration of the distance estimation tracking results of 4 selected trajectories at the tracking rates 24 Hz, and 500 Hz, using tracking modes A3 and B3, compared to the ground truth LiDAR data, as well as the frame-based only approach. All results are presented using YOLOv3 which underperformed the other frame-based object detector but benefitted the most from our proposed event-based methods.

some variance in the trajectory of the vehicle in Figure 4.9 (c) and (d) for both proposed methods at 500 Hz, in contrast with the tracking results at 24 Hz. This variance is caused by the low number of events generated when the vehicle is still after reaching a stop. The lack of motion causes a lower

number of events to be generated. Therefore, the event-based detection and tracking process is affected by the events generated due to noise or the lack of events generated altogether, causing the estimated distance to be unstable, and negatively affecting the correct detection rates. This further highlights the limitations of event cameras in static scenes and stresses the importance of filtering techniques, which are not considered in this study, to improve the event data's signal-to-noise ratio. Likewise, our framework is also affected by a degraded image frame input in conditions such as continuous low-light or motion blur, which the inter-frame event-based detection process is dependent on, at least initially. This case, however, is not considered in this work due to the lack of proper data for evaluation and testing but can be addressed in future works.

In summary, consistent with the results presented in Section 4.3, the tracking configuration mode A3, which entails an event-based mask along with both improvement methods presented, consistently provides the best results under the varying tracking rates and either frame-based object detectors. Overall, we conclude that this experiment is successful. Results show tracking results within the ground truth data's margin of error, at various temporal resolutions and tracking rates. Thus, reaffirming the capabilities and advantages of incorporating event data with proper event-based techniques to improve the performance of a frame-based framework by applying low-cost, classical image processing and computer vision techniques on the asynchronous, and high-temporal-resolution event data.

4.6 Conclusion

In this work, we have presented an improved, high-temporal-resolution object detection and tracking framework using a combination of frame and event-based methods. Building on our prior work [45], we have introduced two event-based methods that further enhance the robustness and accuracy of the detection and tracking framework. These methods are event-based BB refinement and continuous event-based object detection and recovery. Using a labeled MOT vehicle dataset with HOTA metrics, an ablation study was conducted, which showed that the two methods combined

provide significant performance gains outperforming frame-based and prior approaches alike, at tracking rates from 24 to 384 Hz. The results show that these event-based methods, in combination with optimized and real-time object detector models such as YOLOv3, can benefit substantially by incorporating the asynchronous and high-temporal-resolution event data in a multi-modal approach. More specifically, these methods can reduce the effects of intermittent frame-based object detection with various infrequent missed detections, and improve the precision of a detection's BB, with minimal computational overhead. This was demonstrated by the absolute improvements of 11.5% in the DetA metric, and 7.5% in the overall HOTA metric, compared to the single-modal frame-based approach at 24 Hz using YOLOv3. Nevertheless, these approaches are still susceptible to false detections (*i.e.*, false positives) produced by the frame-based object detectors, which are given higher confidence due to the presumed robustness of these models. This is an indirect result of classical methods that are not quite dynamic and require a decent amount of handcrafting. Instead, some learned event-based methods can be explored to replace some of the proposed components of our presented framework with a more dynamic approach toward handling noise or signal degradations resulting from either modality.

Furthermore, a validation experiment was designed and conducted to demonstrate the usefulness of our hybrid framework using real-world values. A high-speed LiDAR was used to collect ground truth distance measurements for vehicle tracking at a 1000 Hz sampling rate. The vehicle detection and tracking results were generated at various temporal rates, including 24 Hz (equal to the framerate of the APS) as well as 50, 100, 200, and 500 Hz high-resolution tracking rates. The tracking results were assessed using different error metrics and overall detection success rates. Results showed that high-temporal-resolution tracking is feasible with output within the ground truth data's margin of error, as well as very high successful detection rates, yielding a true high-resolution tracking output by utilizing event data appropriately.

Overall, this work demonstrates the effectiveness and capabilities of event-based vision, and how well it can complement frame-based vision for different computer vision tasks. The properties of this sensing modality provide great potential that requires proper methods to fully utilize it.

Future work potential includes replacing some of the presented classical and hand-crafted event-based components with learned ones to achieve a more dynamic and robust performance under various challenging scenarios for both modalities, such as non-static scenes with ego-motion for the event-based methods; and low-light or motion blur for the frame-based methods. Nevertheless, that would require larger amounts of labeled event and multi-modal datasets, and possibly other event representations. Furthermore, this work can be tailored to on-vehicle cameras that are essential in autonomous vehicles and automated driving, with further development and applicable datasets. These approaches would require the ability to differentiate between foreground and background events to enable robust object detection and tracking performance.

CHAPTER 5

CSTR: A Compact Spatio-Temporal Representation for Event-Based Vision

Event-based vision is a novel perception modality that offers several advantages, such as high dynamic range and robustness to motion blur. In order to process events in batches and utilize modern computer vision deep-learning architectures, an intermediate representation is required. Nevertheless, constructing an effective batch representation is non-trivial. In this chapter, we propose a novel representation for event-based vision, called the compact spatio-temporal representation (CSTR). The CSTR encodes an event batch's spatial, temporal, and polarity information in a 3-channel image-like format. It achieves this by calculating the mean of the events' timestamps in combination with the event count at each spatial position in the frame. This representation shows robustness to motion-overlapping, high event density, and varying event-batch durations. Due to its compact 3-channel form, the CSTR is directly compatible with modern computer vision architectures, serving as an excellent choice for deploying event-based solutions. In addition, we complement the CSTR with an augmentation framework that introduces randomized training variations to the spatial, temporal, and polarity characteristics of event data. Experimentation over different object and action recognition datasets shows that the CSTR outperforms other representations of similar complexity under a consistent baseline. Further, the CSTR is made more robust and significantly benefits from the proposed augmentation framework, considerably addressing the sparseness in event-based datasets.

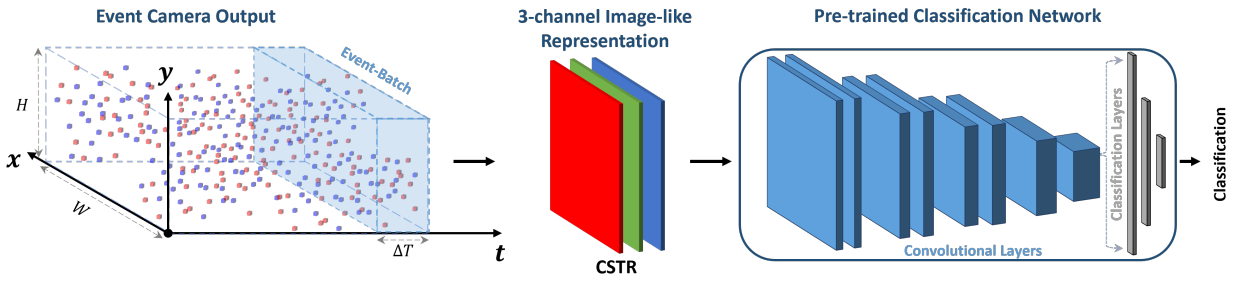


Figure 5.1: Overview of the general framework of this chapter. Sparse and asynchronous events, representing brightness changes at each pixel, are captured using an event-based sensor. To utilize this spatio-temporal event data, an intermediate representation is required to leverage modern deep-learning solutions when processing events in batches. In this work, we propose the Compact Spatio-Temporal Representation (**CSTR**) that encodes spatial, temporal, and polarity information of event data in a 3-channel image-like format. Accordingly, the CSTR is directly compatible with off-the-shelf pre-trained CV architectures.

5.1 Introduction

Perception plays a crucial role in real-time robotic applications, enabling their operation in dynamic and unpredictable environments [13, 20]. These applications often operate under challenging lighting conditions, including high dynamic range (HDR) or high-speed motion scenes. Ensuring accurate perception and prompt responses under such conditions is vital for their success, especially in safety- or time-critical applications like AVs [13] and industrial automation [20]. For instance, in an HDR scene such as when emerging from a tunnel in broad daylight, the failure to detect objects like vehicles or traffic signs can have severe consequences [129]. To address the challenges of robust operation in challenging lighting conditions (*e.g.*, HDR or high-speed motion scenes) and in potentially dynamic and unpredictable environments, many researchers have increasingly turned to event-based vision [29, 55] as a promising alternative visual sensing modality.

Event-based sensors, such as the Dynamic Vision Sensor (DVS) [101] or the Asynchronous Time-Based Image Sensor (ATIS) [140], operate by capturing per-pixel brightness changes asynchronously and at very high temporal resolutions [101, 140]. This results in a spatially sparse yet temporally dense output that effectively represents all visual changes in a scene over a specified time interval. In contrast, traditional cameras capture intensity images at a fixed rate, such as

24 frames per second [21]. This fixed rate can possibly lead to oversampling of static scenes, resulting in redundant data; or undersampling of scenes with high-speed motion, resulting in motion blur [55]. Overall, event-based vision offers several distinct properties that address dynamic range, response time, and motion blur issues. These properties include an HDR of $>$120 dB, microsecond-level temporal resolution, low output latency in the order of microseconds, and low power consumption averaging a few milliwatts [101, 55]. Consequently, these characteristics make event-based vision particularly well-suited for real-time robotic applications [52, 51]. Such applications require accurate perception and prompt response to visual changes, especially in challenging scenarios such as HDR scenes [59], low-light conditions [189], or high-speed motion environments [170]. In comparison, traditional cameras often struggle to perform effectively in such scenarios [129, 51].

While the properties of event-based vision are very compelling, effectively utilizing event data in various applications presents a challenge. The generated event stream is asynchronous and sparse, necessitating its transformation into a compatible format for established algorithmic methodologies. For instance, most traditional object detectors and classifiers employ a three-channel input designed for RGB imagery [149, 72]. However, the independence and sparsity of events make it non-trivial to establish batch relationships, often leading to the creation of hand-crafted representations tailored to specific applications [191, 91]. This inherent problem hampers generalization, as traditional frame-based cameras benefit from standardized formats that facilitate the canonical transfer learning of dataset weights across tasks. In contrast, event-based algorithms, are highly sensitive to the specific type of open-source data and its representation. This further exacerbates the data sparsity issue. As a result, the data needs to be closely associated with the particular task at hand, adversely impacting generalization and posing challenges for training convergence.

Accordingly, most works resort to using image-like representations in order to leverage pre-trained CV models. One common representation is the Event Frame [78, 62], chosen for its simplicity. This representation keeps track of whether any event has occurred at each pixel within a

given time period (where the time period is a variable that can be adjusted per task). By doing so, the batch of events is effectively transformed into a single-channel image (or can be replicated to form a 3-channel image) that can be utilized with existing algorithms. While convenient, this approach has some limitations. Notably, it binarizes the behavior for the specified sampling period, losing temporal and polarity information (brightness changes), and is generally outperformed by more sophisticated approaches [16, 192, 57, 9]. Alternatively, more advanced representations have been explored to capture temporal and polarity contexts [16, 192, 57, 9]. These representations demonstrate better performance, but they come with either the trade-off of notable pre-processing overhead [16, 192] or are not directly compatible with pre-trained CV architectures that require a 3-channel input [57, 9].

To address these challenges, we propose a novel representation for event data called the Compact Spatio-Temporal Representation (CSTR). The CSTR efficiently encodes the spatial, temporal, polarity, and event count information of a given event batch while requiring minimal processing overhead. This is achieved by calculating the mean timestamps of the events per polarity type (positive or negative) and the normalized event counts at every spatial position in the resulting representation frame. This results in a 3-channel image-like format that is directly compatible with existing state-of-the-art networks [149, 72], allowing for seamless integration without the need for additional modifications. We visualize the general framework of this chapter in Fig. 5.1.

We demonstrate the effectiveness of the CSTR through a comprehensive series of well-established event-based recognition benchmarks. This benchmarking includes six well-known representations that are similarly compatible with off-the-shelf networks over the following datasets: N-MNIST [130], N-CARS [161], N-Caltech101 [130], CIFAR10-DVS [93], ASL-DVS [15], and DVS-Gesture [5]. The CSTR is consistently an excellent performer, achieving the highest overall classification accuracy. Furthermore, the CSTR is stable when applying random augmentations; these are demonstrated to notably enhance classification accuracy, validating that the CSTR is a robust approach for encoding event data.

We summarize the contributions of this work as follows:

- We introduce the compact spatio-temporal representation (CSTR) for event-based vision, which efficiently encodes the spatio-temporal information of events in a 3-channel image-like format, directly compatible with modern CV architectures.
- We provide a comprehensive evaluation of the CSTR against foundational event representations of similar complexity using six event-based recognition datasets.
- We propose an augmentation framework for event data, significantly improving the performance of the CSTR and other spatio-temporal representations.
- We demonstrate the effectiveness of the CSTR and the data augmentation framework when combined with off-the-shelf pre-trained classifiers.

The source code developed for this work is available at: github.com/zelshair/cstr-event-vision

5.2 Related Work

Event-based vision has recently seen significant advancements that leverage its unique characteristics for various applications [55, 189, 170, 51, 9]. There are two general approaches to effectively utilize the asynchronous and sparse event data. These include *event-by-event* and *batch* processing. In this section, we provide an overview of the relevant methods of each approach, highlighting their strengths and identifying their limitations. Next, we provide an overview of augmentation methods explored in the literature for enhancing event data. Finally, we introduce the proposed CSTR along with a new augmentation in the context of these limitations, noting how they address some of the remaining challenges.

5.2.1 Event-by-Event Processing

Event-by-event processing methods directly utilize events as they are received [156, 56, 84, 42]. This approach is intuitive and minimizes processing delays. The most prominent methods are

spiking-neural-networks (SNNs) [42, 53, 133, 181, 117]. An SNN is a bio-inspired version of artificial neural networks comprising interconnected neurons. SNNs operate by integrating incoming spikes (events at the input layer) over time. An output spike is generated when the membrane potential of a neuron surpasses a certain threshold causing it to reset. The generated output spikes propagate information to other neurons in deeper layers, connected hierarchically. This neuron-activation threshold enables SNNs to be computationally efficient [181, 117, 137].

Despite the computational efficiency and minimal latency of event-by-event algorithms, they suffer from some limitations. Processing events individually inherently lacks temporal context, necessitating tailored solutions to compensate for the lack of event history [156, 56, 84]. Ironically, this approach can become computationally expensive during periods of high event density. Scenes with significant motion and texture can generate a substantial amount of events per second, requiring a proportional number of operations. As event-based sensors continue to improve their frame resolutions [21, 162], this computational challenge will only intensify. While SNNs somewhat address the latter with their energy-efficient design, they are non-trivial to set up and implement [42, 53, 133]. Moreover, SNNs require specialized hardware, which limits their widespread adoption, posing additional barriers to deployment.

5.2.2 Batch Processing

Batch processing methods accumulate, encode, and classify the events generated in a given time period. These approaches add temporal context with the capability to provide synchronous responses (i.e., a classification per each batch period). By applying an intermediate encoding method, they have the key benefit of being able to employ modern computer-vision networks. This is directly germane to the problem statement of being able to leverage existing state-of-the-art networks (and corresponding training weights). Hence, we focus this survey on event-batch representations that are compatible with frame-based networks.

5.2.2.1 Image-Like Representations

Many opt to represent event batches in a simple image-like format. These representations encode spatial, temporal, and/or polarity information into traditional one, two, or three-channel images. Such approaches are popular because they enable rapid prototyping and demonstrate strong performance across various perception tasks [62, 78, 114]. For example, the Event Frame encodes the event's spatial information (*i.e*. the existence of any events per spatial position) [78], while the Event Count (also known as Event Histograms) [7, 106, 114] indicates the number of events recorded, instead. More advanced versions of these representations incorporate polarity information as well [7, 62, 114]. These representations, however, are inherently limited as they do not capture the temporal information of the event data. To address this limitation, more comprehensive representations have been developed to incorporate spatio-temporal information in an image-like format. One popular representation is Timestamp Images [134], also referred to as Time Surfaces [91]. Timestamp Images encode the timestamp information of the latest event at each spatial index [134], often represented using a separate channel per polarity type resulting in a 2-channel representation [134]. Recent advancements related to Timestamp Images have explored sophisticated techniques to enhance robustness against noise [86, 161]. For instance, DiST [86] incorporates temporal discounting by considering the ρ spatio-temporally neighboring events at each spatial position. Thus, discounting the timestamps of the latest events using a normalized time range of the neighboring pixels.

One challenge encountered in temporal representations is motion overwriting. While timestamp images excel in retaining contour information, the recent timestamps can be overwritten. This can happen when using long batch periods or in highly textured scenes. Accordingly, various representations have emerged that incorporate both the temporal and count information of events in different forms [3, 8, 175, 190]. For instance, a 4-channel representation, known as Event Image [175, 190], incorporates recent timestamps and event count per polarity. Another work by Bai *et al*. [8] proposes a more compact 3D representation that includes the temporal information of both polarities as well as the event count in separate channels. This forms a spatio-temporal

image-like representation that encompasses vital information about the event data. The authors also investigate the advantages of this approach in the context of event-based object recognition.

Overall, the limitation of most spatio-temporal image-like representations can be distilled to overlapping events. A high number of overlapping events often results when using long batch periods or when operating in highly textured scenes. This can result in the overwriting of recent events causing a loss of information. Shortening the batch period can potentially limit this issue [190], however, this reduces temporal context and increases processing frequency.

As an alternative, image reconstruction from events is an effective approach that results in intensity images that enable the direct use of modern frame-based CV architectures [147]. However, generating images from events is a very processing-heavy task, making it not very suitable for real-time systems.

5.2.2.2 Advanced 4D Grid-Like Representations

Advanced grid-like representations have been proposed to overcome the issue of event overlapping, thus, retaining more information [9, 57]. For example, TORE volumes [9] utilizes a first-in-first-out buffer at each spatial position to retain the temporal information of the last K events, for both polarity types, where $K > 1$. This results in a Four-dimensional (4D) representation with a resolution of $2 \times K \times H \times W$, where H and W are the frame's height and width, respectively. By doing so, TORE volumes [9] limit the problem of event-overwriting which is often encountered in image-like representations.

Another notable representation is Event Spike Tensors (EST) [57]. EST employs an end-to-end learning approach to derive event representations from input data. This is achieved by applying convolutional operations on a batch of events with a learned kernel comprising a multi-layer perceptron with two hidden layers. Then, the resulting convolutions are discretized, yielding a 4D grid-like representation with dimensions of $2 \times B \times H \times W$, where B is the pre-selected number of temporal bins.

Although these representations demonstrate remarkable performance in a multitude of tasks

[9, 57], it is important to note that the choice of compatible DL architectures is somewhat limited. Consequently, an additional quantization step is often required to convert the 4D representation into a 3D format [57]. An alternative approach involves splitting the 4D grid along the polarity dimension (first dimension) and employing multiple DL models in parallel to process the resulting outputs, or modifying the input layers of a DL model to accommodate the higher-dimensional input. However, both approaches may lead to higher memory and computational requirements due to the increased dimensionality of the inputs.

5.2.2.3 Voxel Grids

Voxel grids offer a precise means of capturing the spatial and temporal characteristics of events. A voxel represents a 3D point, traditionally denoting the height, width, and depth coordinates in a 3D model. Combining these voxels creates a 3D structure known as a voxel grid. Voxel grids are widely used in 3D CV, especially for representing a LiDAR-generated point cloud [68]. Similarly, it can be also used to handle sparse event data. Voxel grids are applied to event batches by converting the depth axis to a temporal axis using B temporal bins per event batch. This conversion is typically achieved through spatio-temporal quantization employing a designed sampling kernel. The resulting voxel grid has dimensions of $B \times H \times W$, allowing it to retain the essential spatio-temporal relationships within the event batches [148, 191, 192]. Accordingly, researchers have explored the application of voxel grids in various CV tasks, including optical flow estimation [191, 192], HDR video reconstruction [148], and object recognition [179].

Despite their advantages, the use of voxel grids poses two primary challenges. Firstly, generating voxel grids can be computationally demanding, especially when utilizing sophisticated sampling kernels. Secondly, the adoption of voxel grids may lead to high memory requirements due to the resulting increased input dimensionality, similar to the challenges with 4D representations discussed earlier. This issue becomes particularly prominent with high-resolution grids (i.e., a large number of bins B) and long batch periods.

5.2.2.4 Graph-Based Representations

Alternative to voxel-grids, events can be represented as graphs [15, 16, 155]. Here, each sampled event in an event batch is treated as a vertex v_i. These vertices v (also referred to as nodes) are then connected to each other using edges ε, based on a pre-defined spatio-temporal distance metric, forming the graph G. This approach similarly captures the temporal relationships within the event batch and offers compatibility with existing Graph Convolutional Networks (GCNs) [15, 16]. Graph-based solutions provide flexibility in the processing of the event data, allowing for a natural way to incorporate their spatial and temporal information [15, 16, 155]. Compared to traditional CNNs, GCNs exhibit significantly lower inference computational complexity [155].

Nevertheless, generating the graphs can be computationally demanding. This is particularly true when dealing with high-density event streams, resulting in a large number of vertices and edges [150]. Consequently, it is often necessary to sample a subset of events from the batch to reduce storage and computational costs [16, 155]. Moreover, unlike CNNs in traditional CV, there is limited availability of Graph Convolutional Network (GCN) models pre-trained on large-scale datasets. This hampers the ability to leverage transfer learning. As a result, researchers often develop their own GCN architectures to accommodate the generated graphs [15, 16, 155].

5.2.3 Augmentation Methods for Event-Based Vision

Data augmentation techniques play a crucial role in enhancing the performance and generalization of DL models. Given the limited availability of labeled event-based datasets, augmentation methods offer an effective approach to expand the training data and improve model robustness. In this subsection, we provide an overview of the different augmentation methods proposed for event data.

Li *et al*. [98] propose several randomized geometric augmentations for training SNNs. These include common techniques such as horizontal flip, translation, and rotation; as well as other unique techniques such as cutout, shear, and CutMix. These transformations introduce variations and enhance model performance. Gu *et al*. [65] introduce EventDrop, an augmentation framework for

randomly dropping events within an event batch. It explores various event-dropping techniques, including dropping events within a random time period, pixel area, or a random portion of the sampled events. EventDrop improves robustness and has been evaluated for event-based object recognition. The authors also explore the use of EventDrop on different combinations of event representations and pre-trained classification models. EventMix [158] presents an advanced augmentation framework that uses a random 3D mask to mix different event-batch samples and their labels. This mixing technique enhances the diversity of the training data and has been evaluated on a set of event-based recognition benchmarks as well. Naeini *et al*. [124] propose spatial, noise, and time-series augmentations to improve contact-force estimation. Spatial augmentations include rotations and resizing. Noise augmentations add sequences of noise to the dataset, which are generated by recording similar sequences without any movement. Time-series augmentations include frame-shifting, which shifts all generated batch-representation frames within a given sequence; and temporal event shifting, where a fraction of events are randomly selected and removed from one frame and appended to an adjacent frame. For both types of time-series augmentations, the authors explore a fixed index-shift range of $+3$ to -3. These augmentation methods, along with others proposed in the literature, contribute to addressing the dataset scarcity issue in event-based vision. By applying these techniques, models can better handle variations in event data and improve their generalization capabilities. However, despite their importance, event data augmentation techniques are still not thoroughly explored in the literature.

5.2.4 Literature Contribution

In this chapter, we present the CSTR, an alternative image-like representation for event-based vision. The CSTR offers a comprehensive representation of sparse event data when processed in batches while requiring minimal memory resources. It provides a choice that eliminates the need for manual parameter tuning and can be generated in an online manner. It is important to note that the CSTR is not meant to replace advanced or more sophisticated representations. Rather, it serves as an excellent representation choice for initial proof-of-concept and facilitates the rapid

deployment of event-based solutions. This is due to the compact 3-channel image-like format of the CSTR, which enables the direct utilization of state-of-the-art CV architectures.

To validate the effectiveness of the CSTR, we conduct several experiments on various event-based recognition benchmarks comparing it to other image-like representations of similar complexity using various pre-trained classification networks. Additionally, we supplement our representation with several randomized augmentation methods that impact different components of events, including spatial, temporal, and polarity. These augmentation techniques further contribute to improving the performance and the generalization capabilities of event-based vision models.

5.3 Methodology

In this section, we present our proposed event-based representation. First, we provide a detailed overview of how events are generated. Then, we define the common and foundational image-like representations that form the basis of our work. These representations fundamentally encode the spatial and/or temporal components of events within the event batch. By analyzing the characteristics of these representations, we derive a more advanced spatio-temporal representation that enhances performance. We visualize these representations on the evaluation datasets in Fig. 5.2 (see: next page). Given that our approach aims to improve temporal context, we also introduce a novel temporal augmentation technique to address the sparseness of training data.

5.3.1 Event Generation Model

In contrast to traditional cameras, event-based sensors capture per-pixel brightness changes, asynchronously [101]. At a given pixel (x, y), an event e is generated whenever the logarithmic change in brightness intensity exceeds a predefined contrast threshold C. This can be expressed as follows:

$$|log(I(x, y, t)) - log(I(x, y, t - \Delta t))| \geq C, \tag{5.1}$$

where $I(x, y, t)$ represents the intensity measurement at spatial position (x, y) at time t, and Δt represents the time duration since the last generated event at the same spatial position. The polarity p of an event is determined by the sign of the brightness change. A brightness increase (on event) is assigned $p = +1$, while a brightness decrease (off event) is assigned $p = -1$. Thus, $p \in \{+1, -1\}$. Event-based sensors report each captured event e_i as a combination of a microsecond timestamp t_i, a polarity p_i, and a two-dimensional spatial coordinate (x_i, y_i). In general, an event stream ε composed of n sequential events can be denoted as:

$$\varepsilon \rightarrow \{(t_1, x_1, y_1, p_1), (t_2, x_2, y_2, p_2), \ldots, (t_n, x_n, y_n, p_n)\}. \tag{5.2}$$

Events can be grouped into batches either based on a specified batch-sampling period ΔT or a fixed number of events. In this work, we focus on event batches accumulated using predefined batch periods to enable a synchronous response.

The event generation process outlined above captures the spatio-temporal dynamics of the scene. This is done by detecting changes in brightness intensity and encoding them as events with corresponding timestamps, spatial coordinates, and polarities.

5.3.2 Foundational Event Representations

To represent a batch of events ε captured during a sampling period ΔT, several image-like representations can be formed. We identify five foundational approaches identified in the literature: Binary Event Frame, Polarized Event Frame, Binary Event Count, Polarized Event Count, and Timestamp Image. While these representations are not typically referred to as *Binary* or *Polarized*, we use these terms to distinguish between them clearly. We detail these approaches next.

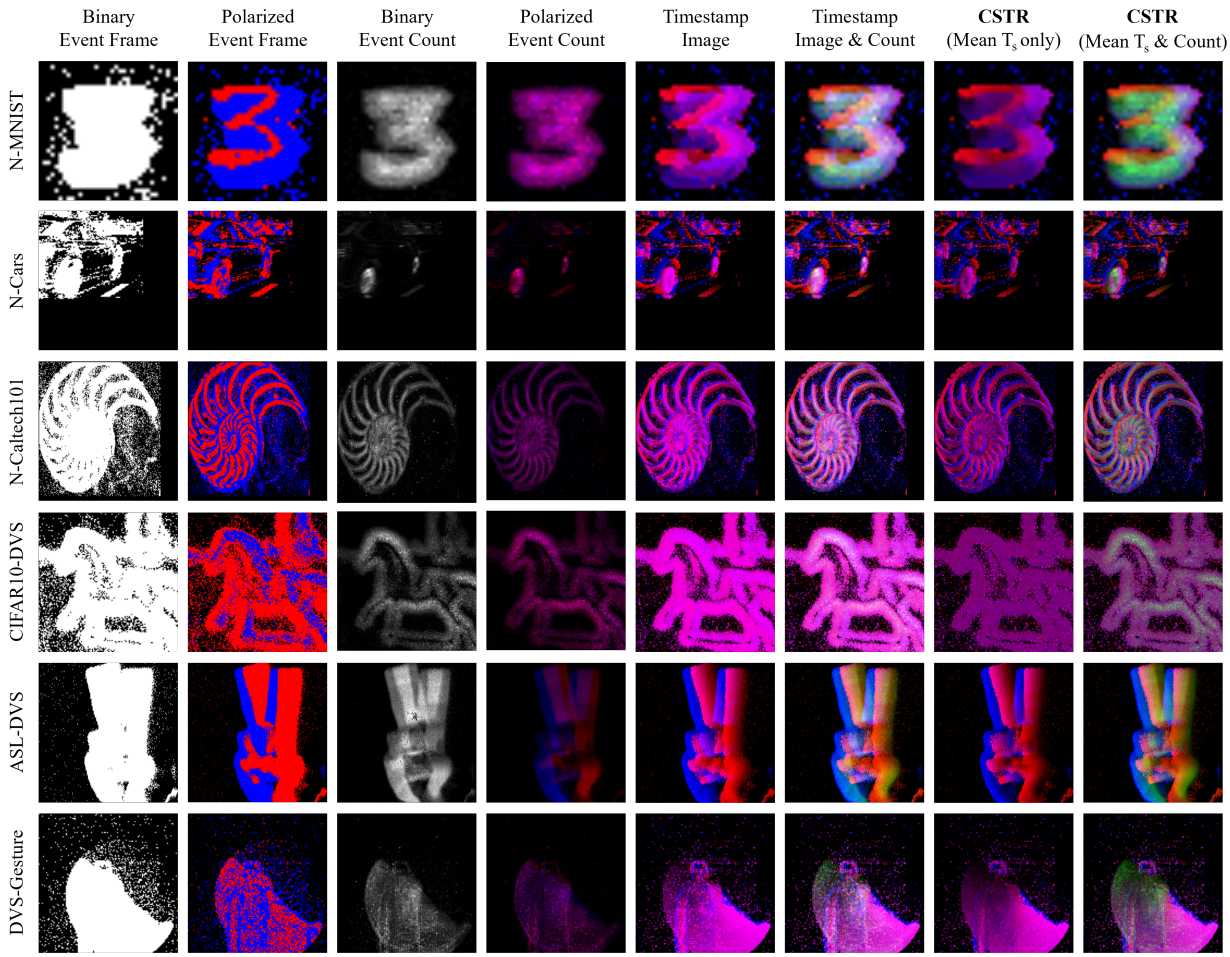


Figure 5.2: Visualizations of the CSTR as well as the foundational event representations investigated in this work using various object and action recognition datasets. To enable visualization, we normalize the Binary and Polarized Event Count representations. Further, due to the significant event noise present in the N-Caltech101 [130] samples, we amplify the event count channels by a factor of 20 to improve visualization. This is shown in the 3rd row, columns 3, 4, 6, and 8.

5.3.2.1 Binary Event Frame

The Binary Event Frame binarizes whether any events are detected at a given spatial location. Each pixel position in the resulting two-dimensional $H \times W$ representation can be encoded as follows:

$$F_{\text{bin}}(x, y) = \begin{cases} 1, & \text{if } x = x_i \ \ \& \ \ y = y_i \\ 0, & \text{otherwise} \end{cases}, \tag{5.3}$$

where x_i and y_i are the spatial coordinates of each event e_i in the batch ε. We encode the presence of an event as 1 and the absence of any as 0. This representation is visualized in Fig. 5.2, column one. Note how this approach is very simplistic and has low contrast; this is because it is highly sensitive to motion-overlapping, where multiple events occur at the same spatial location, as well as noise captured by the event camera. Accordingly, this representation suffers from frame saturation which results under almost any batch-sampling duration, as shown in Fig. 5.2.

5.3.2.2 Polarized Event Frame

The Binary Event Frame can be extended to include polarity information. The Polarized Event Frame incorporates this in a $2\times H\times W$ 3D matrix. The event batches are defined by:

$$F(x,y,p) = \begin{cases} 1, & \text{if } x = x_i \ \ \& \ \ y = y_i \ \ \& \ \ p = p_i \\ 0, & \text{otherwise} \end{cases}, \tag{5.4}$$

where x_i and y_i are the spatial coordinates and p_i is the polarity of each event e_i. We similarly encode detected events by 1 and the absence of events as 0, but for each polarity. This representation is visualized in Fig. 5.2 (second column), showing a notable contrast improvement. Similar to the Binary Event Frame, this representation also suffers from frame saturation. Accordingly, both Event Frame representations are more effective when generating batches based on a constant number of events (ideally a low number) instead of a fixed sampling duration [62].

5.3.2.3 Binary Event count

Alternative to the Binary Event Frame, the Binary Event Count representation captures the number of events at each spatial position. We encode this with the following equation:

$$C_{\text{bin}}(x,y) = \sum_{i=1}^{n} [x = x_i \ \ \& \ \ y = y_i], \tag{5.5}$$

where n is the number of events. The Iverson bracket here would be equal to 1 if the expression is true, which is whenever an event has the same spatial location as the pixel (x, y). This representation retains more information about the scene at each spatial location. Moreover, as visualized in Fig. 5.2 (third column), this representation shows high temporal precision, albeit at the cost of less sharp contour details.

5.3.2.4 Polarized Event Count

Analogous to the Polarized Event Frame, the Binary Event Count can be extended to include event-polarity context. We similarly represent this with a $2 \times H \times W$ matrix as follows:

$$C(x, y, p) = \sum_{i=1}^{n} [x = x_i \ \ \& \ \ y = y_i \ \ \& \ \ p = p_i], \tag{5.6}$$

where n is the number of events, x_i and y_i are the spatial coordinates and p_i is the polarity of each event e_i. This is visualized in Fig. 5.2 (fourth column), improving the contour details (though still not as sharp as the Polarized Event Frame). In contrast to the Event Frame representations, the Binary and Polarized Event Count representations do not suffer from frame saturation. Instead, they are robust to long batch-sampling durations, as shown in Fig. 5.2. Nevertheless, both Event Count representations require significant motion overlap and high event-density streams to yield a meaningful signal.

5.3.2.5 Timestamp Image

An alternative approach to tracking the number of events is to identify the most recent timestamp instead. This is achieved using the Timestamp Image representation [134], which is a 3D matrix of size $2 \times H \times W$. Assuming that the batch's events are sorted in chronological order (*i.e.*, from oldest to newest) we obtain this representation as follows:

$$T_s(x,y,p) = \begin{cases} \frac{t_i - t_s}{\Delta T}, & \text{if } x = x_i \ \ \& \ \ y = y_i \ \ \& \ \ p = p_i \\ 0, & \text{otherwise} \end{cases}, \tag{5.7}$$

where t_s is the raw time offset representing the start of the event batch with temporal duration ΔT, and t_i is the timestamp of the event e_i. In (5.7), $T_s(x,y,p)$ represents the normalized timestamp (in the range of $[0,1]$) of the latest event occurring at the pixel location (x,y) and polarity p. The subtraction of t_s removes the time offset from each event's timestamp. This representation is visualized in Fig. 5.2 (fifth column), where the normalized recent timestamp further improves contour details over the naive Event Frame representations. Note, however, that this improved contrast diminishes under high-density event streams with long batch periods. Additionally, the Timestamp Image is also susceptible to noise in more recent events.

5.3.2.6 Combining Timestamp Image & Event Count

Given the inherent limitations of the Timestamp Image and the Event Count representations, combining them can enhance their robustness [8]. To achieve this, we concatenate the two-channel Timestamp Image T_s, defined in (5.7), with the normalized one-channel Binary Event Count. The normalized Binary Event Count $\hat{C}_{\text{bin}}$ is defined as follows:

$$\hat{C}_{\text{bin}}(x,y) = \frac{C_{\text{bin}}(x,y)}{\max(C_{\text{bin}})}, \tag{5.8}$$

where $\max(C_{\text{bin}})$ is the maximum event count in the frame. This combination results in a $3 \times H \times W$ 3D matrix, as visualized in Fig. 5.2 (sixth column). While the addition of the event-count information improves the contour details, the contrast of the recent timestamp channels is still affected by long batch periods with high event density.

5.3.3 Compact Spatio-Temporal Representation

The combined Timestamp Image and Event Count representation is generally robust but can lose temporal context with motion-overlapping. A recent timestamp is most useful when the event data is temporally sparse; however, can lose general temporal context when there are many overlapping events. This bias can happen frequently when subjected to highly textured scenes or long batch periods. To address this, we introduce the compact spatio-temporal representation (CSTR).

The CSTR improves the timestamp information by utilizing the mean timestamp instead to better capture temporal context. Thus, we initially accumulate the normalized timestamp values of all events at each spatial position as follows:

$$S(x,y,p) = \sum_{i=1}^{n} \begin{cases} \frac{t_i - t_s}{\Delta T} & \text{if } x = x_i \ \ \& \ \ y = y_i \ \ \& \ \ p = p_i \\ 0, & \text{otherwise} \end{cases}, \tag{5.9}$$

where $S(x,y,p)$ represents the sum of the normalized event timestamps at position (x,y,p). Then, we calculate the mean of events' timestamps by dividing (5.9) over (5.6) as follows:

$$\bar{T}_s(x,y,p) = \begin{cases} \frac{S(x,y,p)}{C(x,y,p)}, & \text{if } C(x,y,p) \neq 0 \\ 0, & \text{otherwise} \end{cases}, \tag{5.10}$$

where $\bar{T}_s(x,y,p)$ represents the mean timestamp at position (x,y,p). This is visualized in Fig. 5.2 (seventh column). Nevertheless, mean timestamps on their own can be insufficient to represent the event data. Incorporating the event count can provide vital event-overlap context. Therefore, we concatenate the 2-channel mean timestamp $\bar{T}_s$, defined in (5.10), with the normalized Binary Event Count $\hat{C}_{\text{bin}}$, defined in (5.8). This yields a 3-channel representation. We visualize the CSTR in Fig 5.2 (last column), showing that it retains strong temporal context and contour sharpness. Hence, the CSTR approach adds robustness to motion-overlapping while retaining direct compatibility with existing computer-vision networks.

5.3.4 Event-based Data Augmentation Framework

Randomized data augmentations can improve the generalization of DL models. Further, they can complement the spatio-temporal representations in event-based solutions. Accordingly, we propose a simple framework for randomized event-data augmentations that affect the spatial, temporal, and polarity information of event data. These augmentations can be combined and applied when training an event-based DL model with a spatio-temporal representation.

5.3.4.1 Spatial Augmentations

Spatial augmentations are a common solution for introducing variations across the spatial dimension. In our framework, we explore a combination of rotations, rescalings, crops, and horizontal flips, each with its own parameters to set. For optimal computational efficiency, we apply spatial augmentations to the generated image-like event-batch representations.

5.3.4.2 Temporal Augmentations

Rich temporal information is a major component of event data. Temporal augmentations can help enhance a model's ability to handle temporal dynamics. This is vital for representations that incorporate temporal information (*e.g.*, Timestamp Image [134]). As illustrated in Fig. 5.3, events are shifted based on a randomized value λ within the range of $[-1, +1]$, which is generated per event-batch sample ε. This dynamic but consistent temporal shifting allows the model to learn from different temporal perspectives and improves its robustness to varying temporal dynamics. The temporal shift for each event e_i in the event batch ε can be expressed as:

$$t'_i = t_i + \theta_t(\lambda \Delta T), \tag{5.11}$$

where t'_i is the shifted timestamp of event e_i, θ_t is the max temporal shift threshold ($\theta_t \in (0, 1)$), and ΔT is the batch-sampling period. A balanced value for the max temporal shift threshold θ_t is 0.5, which indicates that the batch's events can be only shifted by a max of $\frac{\Delta T}{2}$ in either direction

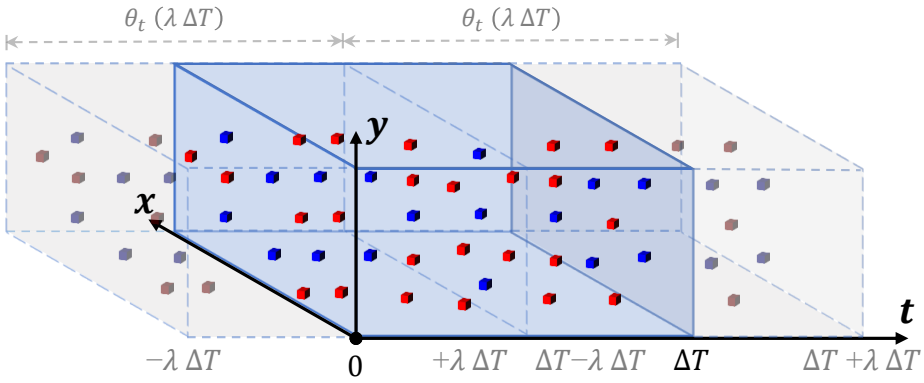


Figure 5.3: Illustration of the proposed temporal augmentation method. Spatio-temporal events within a given batch are uniformly time-shifted by a randomized value λ multiplied by ΔT. Events that fall outside the original temporal range $[0, \Delta T]$ are subsequently removed. The maximum temporal shift θ_t that is demonstrated here is $\pm 50\%$ of the batch duration ΔT.

(shown in Fig. 5.3). Then, we filter out any events that fall outside the original batch's temporal range of $[0, \Delta T]$. Note that the proposed temporal augmentations are applied to a given event batch ε before generating an image-like representation.

5.3.4.3 Polarity Augmentations

Polarity augmentations introduce variations across the polarity domain, enabling the model to learn from varying polarity correlations of events. In our framework, we adopt a simple approach of inverting all the polarities in an event batch prior to frame transformation. This polarity inversion typically implies the reversal of the direction of motion and can introduce robustness to variations in lighting and motion. Hence, for each event e_i in an event batch ε, the polarity p_i is inverted to $\bar{p}_i$ if the threshold θ_p is met. The threshold θ_p is ideally set to 0.5, indicating a 50% chance of inverting the polarities of a given event batch ε. Similar to the proposed temporal augmentation method, the polarity augmentations are applied before generating the image-like representation.

Table 5.1: Statistics of the event-based object and action recognition datasets used in our experiments.

Parameter	Dataset					
	Object Recognition				Action Recognition	
	N-MNIST [130]	N-Cars [161]	N-Caltech101† [130]	CIFAR10-DVS† [93]	ASL-DVS† [15]	DVS-Gesture‡ [5]
Number of classes	10	2	101	10	24	11
Dataset Type	Static	True	Static	Static	True	True
Event Camera / Event Sensor	ATIS [140]	ATIS [140]	ATIS [140]	DVS-128 [101]	DAVIS-240c [21]	DVS-128 [101]
Frame Dimension ($W \times H$)	35×35	128×128	240×180	128×128	240×180	128×128
# Total Samples	70000	24029	8709	10000	100800	38962
# Train Samples	60000 (86%)	15422 (64%)	6967 (80%)	8000 (80%)	80640 (80%)	30,978 (80%)
# Test Samples	10000 (14%)	8607 (36%)	1742 (20%)	2000 (20%)	20160 (20%)	7,984 (20%)
Avg ± Std of # samples/class	7000 ± 399.3	12015 ± 321.5	86 ± 119.3	1000 ± 0.0	4200 ± 0.0	3542 ± 1122
Min-Max range of # samples/class	6313-7877	11693-12336	31-800	1000-1000	4200-4200	2503-6676
Average # events/sample	4176	3966	115298	205072	28149	27339
Average event-batch duration	310 ms	100 ms	300 ms	1298 ms	110 ms	481 ms

† Indicates that the dataset does not have an official test split. ‡ Denotes that the dataset's original sequences were divided into samples of 500 ms with a 250 ms step size (following [173]).

5.4 Experiment Setup

In this section, we evaluate the proposed event-based representation for object and action recognition. Our primary comparison is evaluating our proposed event representation, the CSTR, against the foundational representations defined in the methodology (Section 5.3.2). We do this over a series of well-known datasets to demonstrate our improvements in recognition tasks. Next, we take the best-performing spatio-temporal representations and do a second comparison while employing our proposed augmentation framework. Our experimental setup, including the network structures, datasets, augmentations, and training parameters are introduced next.

5.4.1 Exp I: Baseline Representation Evaluation

In the baseline experiment, we compare the CSTR against the six foundational event representations presented in Section 5.3.2. Recall that the Event Frame representations are traditionally encoded as either 0 or 1, while the foundational Event Count representations are encoded as the number of events (without scaling). However, the Event Count channel associated with the combined Timestamp Image & Event Count and the CSTR is normalized. This is done by dividing each event-count value by the maximum number of events in the frame as defined in (5.8). We apply this because the temporal representations are already scaled to be in the $[0, 1]$ range.

We add rigor by exploring three-channel configurations for the one- and two-channel representations. We do this to enable direct compatibility with the classification networks' input structures and better leverage their pre-trained weights. In the case of the one-channel Binary Event Frame and Binary Event Count, we replicate the resulting channel three times. In the case of the Polarized Event Count, Timestamp Image, and the CSTR with mean timestamps only, we append an empty channel of zeros of the same spatial dimensions. Lastly, for the two-channel Polarized Event Frame, we first convert to an intermediary one-channel representation, where positive and negative events are denoted by values of $+1$ and -1 (following the approach proposed in [78]). We then replicate this three times instead of padding with a channel of zeros. These configurations are determined through experimentation to yield optimal results for each representation.

5.4.1.1 Event-Based Recognition Datasets

Several event-based object and action recognition datasets are available in the literature. In this work, we utilize four commonly used event-based datasets to evaluate our proposed methods for object recognition: N-MNIST [130], N-Cars [161], N-Caltech101 [130], and CIFAR10-DVS [93]. Additionally, we evaluate our methods on two action recognition datasets, namely ASL-DVS [15] and DVS-Gesture [5]. In Table 5.1, we provide an overview of the main details and statistics of the selected recognition datasets.

For object recognition, all datasets except N-Cars [161] are effectively event-based versions of their frame-based counterparts commonly used in conventional CV. These datasets are generated using an event-based sensor, such as the DVS-128 [101] or the ATIS [140], mounted on a platform that moves in parallel to a screen displaying image samples of each dataset. The platform is programmed to move at various velocities and motions to simulate events similar to real-world sensor data. N-Cars [161], on the other hand, was generated using an event camera mounted on a moving vehicle driving on real-world roads. The dataset consists of events captured by the event camera as the vehicle encounters different objects, including cars and pedestrians, in various driving scenarios.

For action recognition, ASL-DVS [15] consists of 24 hand shapes resembling different letters from the American Sign Language. These shapes were recorded in an office environment with constant illumination using DAVIS240c [21]. For each letter, 4200 samples were collected at a sampling duration of 100 ms. Meanwhile, DVS-Gesture [5] consists of 1342 event-data sequence recordings of 11 different gestures. These sequences were captured under three lighting conditions and performed by 29 individuals. Due to the considerable length of the dataset's sequences ($\sim$100 seconds on average), we divide each into shorter samples of a fixed batch-sampling period. Initially, each sequence is split into a subsequence per gesture. Then, the resulting subsequences are further divided into 500 ms samples with a 250 ms step size, following a similar approach used in previous works [173, 179, 16]. The resulting number of samples is presented in Table 5.1.

Except for DVS-Gesture [5], we use the provided samples with pre-defined batch periods ΔT from each dataset, as outlined in Table 5.1. The sampling periods range from 100 ms (N-Cars [161] and ASL-DVS [15]) to roughly 1300 ms (CIFAR10-DVS [93]). This enables us to analyze the robustness of different event representations to various batch-sampling periods.

Furthermore, Table 5.1 demonstrates an uneven distribution in the average number of samples per class across the datasets. N-MNIST [130], N-Cars [161], ASL-DVS [15], and DVS-Gesture [5] exhibit a substantial number of samples per class facilitating effective training and fine-tuning of classifiers. In contrast, CIFAR10-DVS [93] and N-Caltech101 [130] have significantly fewer average numbers of samples per class of 1000 and 81, respectively. While the samples of CIFAR10-DVS [93] are uniformly distributed among classes, the samples N-Caltech101 [130] are highly unbalanced, ranging from 31 to 800 samples per class, posing a challenge for object recognition tasks.

For datasets without an official test split (N-Caltech101 [130], CIFAR10-DVS [93], and ASL-DVS [15]), we adopt the 80%-20% training-testing dataset-split strategy employed in similar works [16, 179, 161]. These splits are generated once and utilized consistently throughout the experiments of this work to ensure consistent benchmarking and fair comparisons. In addition, to address the imbalance in the sample distribution within N-Caltech101 [130], we apply the same split ratios

to each class's samples. This approach avoids imbalanced splits and maintains a fair and consistent benchmarking process across the different methods evaluated in this work.

5.4.1.2 Classification Models

We evaluate each event representation using six popular pre-trained CNN image classifiers. We do this both for completeness and to represent real-world use. These classifiers include: ResNet18 [72], ResNet50 [72], MobileNetV2 [154], both Small and Large variants of MobileNetV3 [75], and InceptionV3 [163] (limited to 3-channel representations only). We initialize all networks with weights pre-trained on ImageNet [40]. Then, we replace the final fully connected layer with a corresponding layer that matches the number of output classes in the utilized dataset. For representations with 1 or 2 channels, we replace the initial input convolutional layers of each CNN classifier with randomized weights to accommodate the desired number of input channels. Subsequently, we fine-tune these networks on the evaluation datasets. Throughout our experiments, we observed that utilizing the frame-based architectures as-is (i.e., for 3-channel representations) yields better results due to more effective fine-tuning. Consequently, whenever possible, we present either a replicated or an extended 3-channel version of all tested representations.

5.4.1.3 Training Parameters

For all models trained in this work, we use the cross-entropy loss with the ADAM [87] optimizer (without weight decay), for up to 50 epochs. We utilize an initial learning rate of 1×10^{-3} for N-MNIST [130], N-Cars [161], and ASL-DVS [15]; and 3×10^{-4} for the more challenging N-Caltech101 [130], CIFAR10-DVS [93], and DVS-Gesture [5]. While more advanced learning rate schedulers can be employed, we avoid them to limit the number of hyper-parameters and simplify the comparison.

During training, each batch-representation sample is initially generated with a resolution matching the spatial dimensions of the utilized dataset (as shown in Table 5.1). The resulting 3D representations are then scaled to 224×224 for all classifiers, except for InceptionV3 [163] which

requires a 3-channel input with the spatial dimensions of 299×299. After rescaling, we apply standardization to the resulting 3D matrices using normalization parameters derived from ImageNet [40] (*i.e.*, mean and standard deviation). Our experiments (using the CSTR with the object recognition datasets) consistently show an average classification accuracy improvement of approximately 5% when utilizing ImageNet normalization parameters. This improvement is observed compared to using each dataset's distribution parameters or when not applying normalization. It can be attributed to the suitability of ImageNet parameters for generalizing image-like representations. This is particularly important given the relatively low number of samples of the event-based datasets used in our experiments, compared to ImageNet [40], making them less optimal for removing input bias through standardization.

Furthermore, we randomly split the training set by 75% for training and 25% for validation. In addition, to ensure proper convergence and robust generalization, the samples of the validation split are randomly selected per each class's number of samples. This ensures a more balanced and well-representing validation set. For all models trained in the baseline experiment, we use early stopping to prevent overfitting. Specifically, we monitor the validation loss during training, and if it does not improve for 10 consecutive epochs, we stop the training early to avoid further overfitting. Afterward, we choose the model with the lowest validation loss that results during training. We follow the same procedure when not utilizing early stopping as well. Finally, we use a batch size of 64 for all the models we train throughout this work.

5.4.2 Exp II: Randomized Event Augmentations

With a baseline established, our next experiment aims to leverage the randomized augmentation framework introduced in Section 5.3.4. Augmentations are a popular method for addressing data sparsity as they introduce variance in the spatial, temporal, and/or polarity characteristics. We believe these effects can also be used to further investigate batch-representation stability and explore how well the performance of spatio-temporal representations scales with the proposed randomized event-based augmentation framework.

In this experiment, we explore different settings for each type of randomized augmentation (spatial, temporal, and polarity). For spatial augmentations, we apply crops, rotations, and translations to the generated image-like representations. Initially, we randomly take crops of 90-100% of the spatial frame size with aspect ratios ranging from 3:4 to 4:3. We also apply translations of up to 10% in the x and y axis (up to 5% for N-Cars [161]) and rotations of up to ±10° (up to ±30° for N-MNIST [130]). Additionally, random horizontal flips are used with CIFAR10-DVS [93] (applied prior to the other spatial transformations) with a threshold of 0.5. For both temporal and polarity augmentations, we utilize a balanced value of 0.5 for both the maximum temporal shift θ_t and the polarity inversion thresholds θ_p. We note that all of the proposed randomized augmentations are only applied to the training splits (*i.e.*, excluding validation splits).

Furthermore, we explore different combinations of the proposed augmentation methods. Spatial augmentations can be highly beneficial as spatial dependencies are typically the most informative, especially when identifying the edges or contours of an object. However, when utilizing event data, they require careful manual tuning. On the other hand, the proposed temporal and polarity augmentations have minimal parameters to tune and can naturally complement the training of any event-based solution. Therefore, we focus on the temporal-polarity augmentation combination as an alternative that requires no tuning when using their default threshold values. Finally, for a more comprehensive approach, we explore a combination that incorporates all three event-based augmentation methods.

We perform this experiment only on the spatio-temporal representations presented in this work. This includes the proposed 3-channel variants of the CSTR and the Timestamp Image. These representations are selected because the proposed framework primarily affects the temporal and polarity information of event data, making them optimal for spatio-temporal representations. Additionally, we only utilize the three best classifiers found during the baseline experiment: ResNet18 [72], ResNet50 [72], and InceptionV3 [163]. The ASL-DVS [15] dataset is excluded from this experiment as its performance is already effectively saturated without the use of augmentations. Finally, we provide sufficient training time to ensure reaching an optimal global minimum, by training each

Table 5.2: Average test classification accuracy results for the foundational event representations and the CSTR across different recognition datasets. Each result is the average of up to 6 classification models as specified in Section 5.4.1.2.

Event Representation	**Representation Components**				**Dataset**						**AVG.**
	Timestamp	Polarity	Count	# Channels	N-MNIST	N-Cars	N-Caltech101	CIFAR10-DVS	ASL-DVS	DVS-Gesture	
Binary Event Frame	×	×	×	1	95.1%	91.7%	68.5%	50.6%	<u>99.6%</u>	83.2%	81.4%
				3*	95.2%	92.6%	73.5%	52.8%	**99.7%**	84.2%	83.0%
Polarized Event Frame	×	✓	×	2	96.1%	88.4%	69.8%	62.0%	**99.7%**	90.6%	84.4%
				3*	98.9%	93.2%	81.5%	60.7%	**99.7%**	90.6%	87.4%
Binary Event Count	×	×	✓	1	98.6%	91.6%	75.2%	69.2%	45.8%	87.6%	78.0%
				3*	98.5%	91.1%	81.1%	**73.7%**	78.3%	89.1%	85.3%
Polarized Event Count	×	✓	✓	2	98.9%	91.8%	73.0%	69.9%	41.0%	91.8%	77.7%
				3*	98.5%	92.8%	81.6%	71.8%	52.0%	90.9%	81.3%
Timestamp Image	✓	✓	×	2	<u>99.0%</u>	85.5%	74.1%	67.7%	99.5%	91.4%	86.2%
				3*	<u>99.0%</u>	92.3%	81.3%	68.8%	**99.7%**	<u>93.2%</u>	89.0%
Timestamp Image & Count	✓	✓	✓	3	98.9%	92.2%	82.5%	<u>72.6%</u>	**99.7%**	92.9%	<u>89.8%</u>
CSTR (mean $\bar{T}_s$ only)	✓	✓	×	2	98.9%	92.4%	76.9%	63.5%	<u>99.6%</u>	92.8%	87.4%
				3*	<u>99.0%</u>	92.4%	**83.9%**	67.4%	<u>99.6%</u>	**93.6%**	89.3%
CSTR (mean $\bar{T}_s$ & Count)	✓	✓	✓	3	**99.1%**	**93.6%**	<u>82.9%</u>	71.6%	**99.7%**	**93.6%**	**90.1%**

* denotes that the 1 and 2-channel representations are additionally transformed into 3 channels as specified in Section 5.4.1.
The best and second-best results are highlighted in **bold** and <u>underlined</u>, respectively.

model for 50 epochs without early stopping. We use an initial learning rate of 1×10^{-4} instead while keeping all the other evaluation parameters identical to the initial experiment.

5.5 Evaluation Results

In this section, we present our experimental results. We first do a baseline evaluation of the CSTR and six foundational representations across popular event-based recognition datasets. We then identify the best performers and re-evaluate them when using the proposed augmentation framework. These experiments help show that the proposed CSTR is a robust means of representing event batches, including ones with long temporal durations and high event density. Finally, we present a comparison with other works in the literature.

5.5.1 Exp I: Baseline Evaluation Results

We present the baseline evaluation results in Table 5.2. This table shows the average performance of the representations with all six classification networks detailed in the Experimentation Setup (Section 5.4.1.2). We provide a full breakdown of each network's performance in Table C.1 of

Appendix C. We note a few basic observations. First, including polarity improves generalization. We see this mainly in the Event Frame representations, as well as the Event Count representations but to a lesser extent. This aligns with the methodology expectations. Second, there is a benefit to maintaining the classification networks' native input structure. In all cases, transforming a one or two-channel representation into three channels (by either padding or replicating data) consistently improves classification accuracy. This reinforces the value of transfer-learning frame-based networks for event-based applications. Lastly, our representation, the CSTR, has the highest average classification accuracy and is the best overall in four of the six datasets.

The strength of the CSTR is in addressing motion-overlapping. We can see that of the foundational event representations, the simple Binary Event Count is rather robust. This implies that the number of events per batch is strongly correlated with the classification task, where adding polarity helps better describe the type of motion. Intuitively, this implies that better describing the event's temporal distribution should improve performance. While the Timestamp Image does this via recent timestamps, this approach can be biased for longer temporal periods. The CSTR addresses this by representing the aggregate behavior with the mean timestamp and generalizes very well across datasets, including those with long temporal durations and high event density.

We note the results get particularly interesting with the CIFAR10-DVS [93] dataset. In general, all classification networks for all representations notably overfit. This overfitting concern is verified by the simple Binary Event Count having the highest dataset classification accuracy, remaining in line with its accuracy on other datasets. We believe this overfitting is partially due to the dataset being generated by repeated back-and-forth motions (frequent direction change), causing very significant motion overlap [93]. Furthermore, the CIFAR10-DVS [93] data collection methodology uses up-scaled 32×32 RGB images that appear rather blurry [93]. This blurriness reduces the edge features the events depend on and inherently increases sensitivity to sensor noise. With this said the CSTR still does relatively well, but incrementally worse than the Timestamp Image representations. We hypothesize here that the timestamp recency better correlates with back-and-forth motions versus the timestamp mean.

Lastly, we observe that the optimal classification network can vary across representations and datasets. Intuitively, classification network accuracy should correlate with ImageNet accuracy; however, the expanded results given in Appendix C (Table C.1) show that this is not always the case. We conjecture that this can be a function of dataset density and intra-class variance. When the variance is particularly high, such as in the CIFAR10-DVS [93] dataset, the smaller networks tend to generalize better. This is likely a result of overfitting, where the smaller parameter spaces inherently regularize themselves. However, we also note the large InceptionV3 [163] network is still the top performer for some representations. This implies picking the optimal network may ultimately require experimentation. We recommend that the developer assess various networks and select the one that best fits their accuracy and run-time requirements.

5.5.2 Exp II: Randomized Augmentations Results

We present the results of the augmentation-framework evaluation in Table 5.3. We also provide a detailed breakdown of each network's performance in Table C.2 of Appendix C. Starting with the baseline results, we observe that the CSTR consistently outperforms other representations when considering the top-3 classifiers (ResNet18, ResNet50, InceptionV3) on most datasets. This emphasizes the robustness of the CSTR in capturing spatio-temporal information across varying batch periods. The slight underperformance of the CSTR on the N-Cars dataset compared to the Timestamp Image representation can be attributed to the dataset's low event density and short batch periods. This causes larger classification networks to underfit with more complex representations. We observe this with DVS-Gesture as well. Nevertheless, the introduction of the proposed augmentations highlights the limitations of the Timestamp Image. Specifically, the CSTR demonstrates superior results on N-Cars when utilizing either the temporal-polarity augmentation combination or combining all three augmentation methods. This highlights the CSTR's ability to encode spatio-temporal information optimally when provided with sufficient training variations.

Overall, the augmentation framework shows significant performance improvements across all benchmarks. When using a single augmentation method, the proposed temporal augmentation

Table 5.3: The effects of the proposed event-based augmentation framework on the average test classification performance of the different spatio-temporal representations explored in this work. Each result represents the average classification accuracy of the top three classifiers only (ResNet18, ResNet50, and InceptionV3) due to the complexity of training with augmentations. The first row represents the baseline results obtained without any augmentation, serving as a reference point for each representation. The subsequent rows demonstrate the performance improvements achieved when using the respective augmentation configurations.

Representation	**Augmentation Type**			**Dataset**					**AVG.**
	Spatial	Temporal	Polarity	N-MNIST	N-Cars	N-Caltech101	CIFAR10-DVS	DVS-Gesture	
		Baseline		99.1%	93.4%†	82.0%	72.1%	93.7%†	88.1%
Timestamp Image*	✓			99.3% (+0.2%)	94.5% (+1.1%)	84.4% (+2.4%)	77.5% (+5.4%)	94.1% (+0.4%)	90.0% (+1.9%)
		✓		99.2% (+0.1%)	95.7% (+2.3%)	87.1% (+5.1%)	76.1% (+4.0%)	94.3% (+0.6%)	90.5% (+2.4%)
			✓	99.1% (+0.0%)	95.6% (+2.2%)	86.2% (+4.2%)	71.8% (−0.3%)	93.8% (+0.1%)	89.3% (+1.2%)
		✓	✓	99.2% (+0.1%)	95.8% (+2.4%)	86.9% (+4.9%)	76.3% (+4.2%)	93.9% (+0.2%)	90.4% (+2.3%)
	✓	✓	✓	99.2% (+0.1%)	96.3% (+2.9%)	85.2% (+3.2%)	78.0% (+5.9%)	94.9% (+1.2%)	90.7% (+2.6%)
		Baseline		99.1%	93.3%	84.5%	75.5%	93.0%	89.1%
Timestamp Image & Count	✓			**99.4%** (+0.3%)	95.7% (+2.4%)	84.4% (−0.2%)	**80.4%** (+4.9%)	94.6% (+1.6%)	90.9% (+1.8%)
		✓		99.2% (+0.1%)	95.4% (+2.1%)	87.1% (+2.6%)	77.2% (+1.7%)	94.6% (+1.6%)	90.7% (+1.6%)
			✓	99.2% (+0.1%)	96.3% (+3.0%)	86.4% (+1.9%)	73.5% (−2.0%)	93.3% (+0.3%)	89.8% (+0.7%)
		✓	✓	99.3% (+0.2%)	95.7% (+2.4%)	87.3% (+2.8%)	78.1% (+2.6%)	94.4% (+1.4%)	91.0% (+1.9%)
	✓	✓	✓	99.3% (+0.2%)	96.3% (+3.0%)	87.1% (+2.6%)	80.2% (+4.7%)	94.4% (+1.4%)	91.5% (+2.4%)
		Baseline		99.2%†	92.7%	84.6%	71.5%	93.5%	88.3%
CSTR (mean $\bar{T}_s$ only)*	✓			**99.4%** (+0.2%)	96.1% (+3.4%)	85.7% (+1.1%)	75.6% (+4.1%)	95.5% (+2.0%)	90.4% (+2.1%)
		✓		99.3% (+0.1%)	93.3% (+0.6%)	87.8% (+3.2%)	75.5% (+4.0%)	93.4% (−0.1%)	89.8% (+1.5%)
			✓	99.4% (+0.2%)	96.2% (+3.5%)	87.5% (+2.9%)	70.8% (−0.7%)	94.8% (+1.3%)	89.7% (+1.4%)
		✓	✓	99.2% (+0.0%)	96.9% (+4.2%)	88.3% (+3.7%)	74.8% (+3.3%)	93.7% (+0.2%)	90.6% (+2.3%)
	✓	✓	✓	99.3% (+0.1%)	96.6% (+3.9%)	86.4% (+1.8%)	78.2% (+6.7%)	95.0% (+1.5%)	91.1% (+2.8%)
		Baseline		99.2%†	93.0%	84.9%†	75.8%†	93.4%	89.2%†
CSTR (mean $\bar{T}_s$ & Count)	✓			**99.4%** (+0.2%)	96.3% (+3.3%)	85.0% (+0.1%)	79.3% (+3.5%)	**95.7%**(+2.3%)	91.1% (+1.9%)
		✓		**99.4%** (+0.2%)	95.4% (+2.4%)	87.9% (+3.0%)	78.4% (+2.6%)	94.9% (+1.5%)	91.2% (+2.0%)
			✓	99.3% (+0.1%)	96.1% (+3.1%)	87.0% (+2.1%)	72.2% (−3.6%)	95.1% (+1.7%)	89.9% (+0.7%)
		✓	✓	**99.4%** (+0.2%)	96.6% (+3.6%)	**88.4%** (+3.5%)	77.9% (+2.1%)	94.4% (+1.0%)	91.3% (+2.1%)
	✓	✓	✓	99.3% (+0.1%)	**97.0%** (+4.0%)	86.1% (+1.2%)	79.8% (+4.0%)	**95.7%** (+2.3%)	**91.6%** (+2.4%)

* Indicates that only the augmented three-channel representation versions are considered. † Indicates the best-performing baseline representation. The best and second-best results when incorporating augmentations are highlighted in **bold** and underlined, respectively.

method can match and even exceed the performance of hand-crafted spatial augmentations. This is evident in the highest average performance achieved by a single augmentation method (*i.e.*, 91.2% when using the CSTR). We find that the CSTR benefits the most from the temporal augmentations due to its effectiveness at encoding temporal information. On the other hand, spatial augmentations, while generally reliable, have limitations on datasets with challenging spatial characteristics like N-Caltech101 [130]. Furthermore, spatial augmentations require manual tuning for optimal results. In contrast, the proposed temporal and polarity augmentations serve as a promising alternative, requiring minimal tuning and consistently outperforming spatial augmentations on average across all evaluated representations. This makes them particularly advantageous for optimizing DL models in event-based applications.

Interestingly, we find that combining all augmentation methods (spatial, temporal, and polarity) does not consistently yield the best performance. The significant variations introduced by this combination can lead to underfitting, considering the utilized regularization approach. Therefore, we suggest exploring an alternative approach of randomly selecting one of the augmentation methods per event-batch sample during training. Additionally, we observe that spatial augmentations underperform polarity and temporal augmentations on the N-Caltech101 [130] dataset. This can be attributed to the dataset's imbalance, where typical spatial augmentations are insufficient to improve generalization.

In conclusion, our findings demonstrate the strength of the CSTR and its ability to leverage the proposed augmentation framework. The temporal augmentations prove to be the most advantageous on average for the CSTR, showcasing the CSTR's effectiveness in capturing temporal information. Moreover, combining multiple augmentation methods can enhance generalization performance. However, further exploration and optimization of the augmentation methods are necessary to maximize performance and address limitations.

5.5.3 Comparison with the state-of-the-art

In this section, we compare the performance of the CSTR with other approaches that utilize the same recognition datasets. Although each approach utilizes different methods and training configurations, our aim here is to highlight the efficacy of the CSTR when combined with off-the-shelf pre-trained classification networks. Furthermore, we emphasize how the performance can be further improved by leveraging the proposed augmentation framework for event data.

We present the performance comparison in Table 5.4. While most works report results for an 80-20% split, we provide the results of our framework on a 90-10% split for CIFAR10-DVS [93] as well to establish a fair comparison with those that utilize such a split. For our results on DVS-Gesture [5], we adopt a simple moving-majority filter to handle the long-term temporal dependencies, as applied in [9, 173]. This filter outputs the most frequent gesture classification out of the last 5 (*i.e.*, 1250 ms moving window). If there is more than one gesture with the same

Table 5.4: Comparison with the self-reported state-of-the-art works. Our proposed representation, the CSTR, yields very competitive results when compared with state-of-the-art event-based object and action recognition on the utilized datasets.

Event Representation	Classifier Architecture	Data Augmentation	Dataset					
			N-MNIST	N-Cars	N-Caltech101	CIFAR10-DVS	ASL-DVS	DVS-Gesture
HATS [161]	SVM	×	99.1%	90.2%	64.2%	52.4%	-	-
Event-by-event [53]	SNN	×	**99.6%**	-	-	69.0%	-	96.5%
Graphs [16]	Residual-GCN	✓(spatial)	99.0%	91.4%	65.7%	54.0%	90.10%	97.2%
Graphs [155]	GCN	×	-	94.5%	66.8%	-	-	-
Voxel-grid [179]	GCN	×	99.5%	93.2%	77.8%	69.0%	98.90%	**97.5%**
Event Clouds [173]	PointNet++	×	-	-	-	-	-	95.3%
EST [57]	CNN (ResNet34)	×	-	92.5%	81.7%	-	-	-
Timestamp Image & Count [8]	CNN (ResNet34)	×	**99.6%**	97.3%	89.2%	76.3%	-	-
TORE Volumes [9]	CNN (2×GoogLeNet)	×	99.4%	**97.7%**	83.4%	-	99.95%	96.2%
EST [57]	CNN (ResNet34)	EventDrop [65]	-	95.5%	85.2%	-	-	-
Polarized Event Count [158]	CNN (ResNet18)	EventMix [158]	-	-	84.7%†	84.4%†	-	89.5%
Polarized Event Count [158]	CNN (ResNet34)	EventMix [158]	-	96.6%	89.2%†	**85.6%**†	-	91.8%
CSTR (ours)	CNN (ResNet18)	×	99.1%	93.0%	81.6%	77.8% | 80.6%†	99.88%	95.5%
		TP	99.3%	96.6%	86.7%	77.9% | 81.8%†	**99.98%**	95.5%
		STP	99.3%	96.9%	84.0%	78.8% | 80.9%†	99.44%	96.9%
	CNN (ResNet50)	×	99.2%	92.5%	85.4%	70.6% | 70.4%†	99.94%	97.0%
		TP	99.4%	96.2%	88.6%	75.4% | 77.4%†	99.89%	**97.5%**
		STP	99.5%	96.9%	86.2%	78.7% | 80.9%†	99.84%	96.9%
	CNN (InceptionV3)	×	99.2%	93.5%	87.7%	79.0% | 77.2%†	99.89%	95.9%
		TP	99.4%	96.9%	**89.8%**	80.4% | 83.1%†	99.93%	96.3%
		STP	99.3%	97.2%	88.2%	**81.8%** | 83.7%†	99.74%	**97.5%**

† symbol denotes that the referenced result was based on a 90%-10% split, compared to the typical 80%-20% split (for datasets without an official split). The best and second-best results are highlighted in **bold** and underlined, respectively.

number of classifications (or none), the filter simply returns the classification result for the current event batch. It is worth noting that all the referenced works also utilize a 500 ms sampling period for splitting the event sequences of the DVS-Gesture [5] dataset.

Overall, the results show that the CSTR performs excellently across the employed benchmark datasets. In terms of the baseline performance (excluding augmentations), the CSTR notably achieves state-of-the-art results on CIFAR10-DVS [93] and consistently ranks as the second-best on ASL-DVS [15]. This demonstrates the robustness and versatility of the CSTR which requires minimal configuration and enables a direct and effective deployment for event-based solutions.

To demonstrate the impact of the proposed augmentation framework, we compare the results with other works that incorporate different augmentation techniques for event data. One such work utilizes EventMix [158] augmentations in combination with the Polarized Event Count representation. This work splits the provided batch samples of the N-Caltech101 and CIFAR10-DVS datasets into 10 slices of equal temporal duration. This effectively yields 10 times the original number of samples of each dataset. In contrast, we utilize the provided batch samples of each dataset as-is.

Despite this, the CSTR with the randomized Temporal-Polarity augmentations proves to be highly competitive, even without splitting the datasets' samples. Accordingly, the CSTR demonstrates significant robustness to varying batch periods. Furthermore, we show that the CSTR, in combination with the proposed temporal and polarity augmentations, can achieve stronger results on N-Caltech101 [130] even with less training data. Lastly, the addition of the augmentation framework significantly improves the performance of the CSTR, surpassing more advanced representations such as EST [57] with the EventDrop [65] augmentation framework.

Our findings highlight the strength of the CSTR representation when combined with off-the-shelf pre-trained classifiers. They showcase the effectiveness of the CSTR in capturing temporal information and leveraging the robustness of pre-trained networks without any modification to the input layers. Thus, the CSTR retains a compact input dimensionality and effectively leverages transfer learning. Furthermore, the proposed augmentation framework offers a promising alternative for enhancing generalization performance without the need for significant manual tuning. Finally, we note that the results presented utilize a simple training framework. Therefore, various training optimization and batch-sampling techniques can be explored to further improve robustness.

5.6 Conclusion

In this chapter, we introduce the compact spatio-temporal representation (CSTR) for event-based vision. When dealing with asynchronous event data, it is common to accumulate events in batches to generate a synchronous response. In order to do so, an intermediate representation is needed, especially when utilizing modern CV architectures. Thus, encoding the data into a representation compatible with existing classification networks is crucial for leveraging transfer learning and avoiding the complexity of designing custom deep-learning architectures. Foundational event representations typically encode either the number of events or the most recent event's timestamp per spatial location (based on polarity). These approaches are convenient and relatively robust but can

be sensitive to motion-overlapping (common in long sampling duration) and possibly deficient for high event-density streams.

The CSTR improves upon the foundational event representations by better describing the temporal behavior of the asynchronous event data while retaining similar computational complexity. This is done by calculating the average of the normalized timestamps per each event polarity, combined with the polarity-agnostic number of events at each spatial index of the frame. Besides, the CSTR imposes minimal processing overhead given that each event is only processed once and that each spatial position is updated independently (*i.e.*, without the need to maintain any spatial dependencies), as indicated in the methodology. Accordingly, the CSTR generates a compact image-like representation that is more robust to high-motion scenes and long temporal durations. We validate this hypothesis through rigorous benchmarking against similar representations.

Combining the CSTR with off-the-shelf pre-trained classifiers demonstrates its ability to effectively leverage the power of transfer learning without modifying the input layers, thereby retaining its compact input dimensionality. We also propose a simple yet effective augmentation framework for event data, significantly improving the performance and generalization capabilities of the CSTR. This framework highlights the potential of augmentations in event-based recognition without the need for extensive manual tuning.

Experimental validation confirms that the CSTR outperforms foundational event representations in popular event-based applications. Benchmarking the CSTR against six foundational representations and six common recognition datasets (using six popular classification networks) consistently shows its superior performance. Additionally, incorporating random augmentations during training, including our proposed temporal augmentation, further enhances results on all representations, with the CSTR generally benefiting the most from the proposed augmentation framework. This overall improvement validates the CSTR's ability to robustly encode temporal information.

The CSTR achieves our goal of providing a robust event-batch representation that is directly compatible with existing CV architectures, maintaining similar inference complexity. As a result, the CSTR is an excellent choice for developing event-based solutions. The combination of the

CSTR with the proposed augmentation framework further enhances its performance and generalization capabilities, requiring minimal tuning and enabling direct deployment.

While the CSTR excels as a versatile representation, it does not directly address certain prominent challenges in event-based vision, such as sensor noise [86]. To mitigate these issues effectively, additional techniques may be necessary.

Future work involves exploring the use of the CSTR in other perception tasks, such as object detection, and investigating additional optimization techniques to enhance robustness. Additionally, evaluating the suitability of the CSTR for real-time applications, where latency is a primary concern, would be an interesting avenue to explore.

CHAPTER 6

Exploring Image-like Representations for Event-Based Object Detection

This chapter investigates the application of event-based vision for object detection, focusing on the integration and efficacy of image-like representations, such as the Compact Spatio-Temporal Representation (CSTR), alongside traditional frame-based methodologies. Building on the premise that frame-based object detection methods excel in texture-rich and static environments, this study acknowledges their limitations in dynamic lighting and high-motion scenarios. To address these challenges, it explores the utility of event-based sensors, which are known for their high dynamic range capabilities and immunity to motion blur. Utilizing pre-trained single-stage object detectors and a novel data augmentation framework, this chapter evaluates the CSTR against other image-like event representations and conventional frame-based approaches. The research further extends to multi-modal object detection, aiming to leverage the complementary strengths of event-based and frame-based data to achieve enhanced detection robustness.

6.1 Introduction

Object detection is a fundamental task in the field of CV with significant importance and wide-ranging applications [81, 166]. It is defined as the process of identifying and localizing objects within a scene and classifying them into various categories. Its importance lies in its ability to provide detailed scene understanding, which is crucial for various practical and real-world ap-

plications. Object detection algorithms are integral to numerous applications, including but not limited to, security and surveillance systems [113, 145, 174], medical imaging [4, 176], image retrieval [33, 97], ADAS [85, 138], and AVs [54, 110, 171]. In the domain of autonomous systems, object detection is particularly vital; it underpins the perception capabilities of autonomous vehicles, enabling them to navigate safely by accurately identifying and responding to other vehicles, pedestrians, and obstacles in their environment. This capability is not only essential for ensuring the safety and reliability of such systems but also for advancing their intelligence and operational efficiency.

Recently, object detection methodologies have evolved significantly. Initially dominated by CNN-based two-stage detectors like Faster-RCNN [152], these methodologies were followed by more efficient one-stage detectors such as YOLOv3 [149] and SSD [108]. In our work, we specifically focus on leveraging one-stage detectors due to their balance between accuracy and computational efficiency, which is crucial for real-time applications in event-based vision.

While traditional object detection techniques predominantly utilize frame-based vision systems for their rich texture output, these systems exhibit notable limitations in dynamic lighting conditions, including low-light and high dynamic range (HDR) scenarios, as well as in the presence of substantial motion and adverse weather conditions [70, 73, 109, 116, 126]. Alternative sensing modalities such as LiDAR and radar bolster the perception stack [10, 99, 184] by offering depth estimation at varying spatial resolutions enabling their ability to detect objects, yet they fall short in object classification due to inherent sparsity of their output data, particularly at extended ranges [90].

Event-based sensors present a novel paradigm by asynchronously capturing pixel intensity changes [21, 101, 140]. They excel over traditional frame-based systems by offering HDR output and being inherently immune to motion blur [55], suggesting their potential integration into a robust perception framework. However, the asynchronous and sparse nature of the data generated by event-based sensors poses a challenge for integration with established CV architectures, which are accustomed to synchronous, dense inputs. This challenge is compounded by the scarcity of

labeled datasets that can facilitate broad adoption and utilization of event-based vision.

Building upon the foundations laid in Chapter 5, this chapter explores the application of the Compact Spatio-Temporal Representation (CSTR) within the domain of object detection. We assess the CSTR's performance in translating the asynchronous event data stream into a format compatible with conventional CV algorithms, comparing it against other image-like event representations. Moreover, we compare its efficacy with that of traditional frame-based methods under varied environmental conditions presented in the evaluation datasets. Our exploration also extends to the field of multi-modal object detection, where we aim to utilize the complementary strengths of event-based and frame-based data, aiming to overcome their respective limitations and enhance overall detection robustness.

In this chapter, we aim to make the following contributions:

- A comprehensive evaluation of the CSTR in the context of object detection, highlighting its advantages and limitations compared to other image-like event representations and traditional frame-based approaches.
- An in-depth investigation into the integration of the CSTR and other image-like representations within a multi-modal detection framework using different fusion methods, highlighting potential synergies between event-based and frame-based sensors.
- The development and assessment of a novel data augmentation framework, specifically tailored to event-based and multi-modal data to enhance object detection performance.

6.2 Related Work

6.2.1 Evolution of Object Detection Architectures

Object detection has evolved significantly with the advent of CNN-based DL models. Initially, two-stage detectors such as Faster-RCNN [152], achieved state-of-the-art performance by generating region proposals which are then classified [64, 63, 152]. These were followed by one-stage

detectors, such as YOLOv3 [149] and SSD [108], which streamlined the process at the cost of some accuracy for increased speed [149, 108, 102, 151]. One-stage detectors rely on a pre-defined set of anchors (indicating possible object positions) to perform end-to-end learning. In general, both architectures rely on deep CNN-based encoders, often pre-trained on extensive datasets like ImageNet [40], to extract high-level features that guide the detection process [160, 72].

More recent developments have seen the introduction of Transformer-based models, such as DETR [27], proposing an approach to object detection. These models replace conventional CNN architectures with self-attention mechanisms [169, 43], allowing for end-to-end processing of images and eliminating the reliance on hand-crafted components such as Non-Maximum Suppression (NMS) or anchor generation [27]. However, the extensive training data requirements and slower processing speeds of Transformer-based models present challenges for event-based vision applications, where data is often limited and real-time processing is essential.

In light of these considerations, our research opts for one-stage detectors pre-trained on frame-based data. This choice is driven by the need to balance accuracy with computational efficiency, crucial for real-time applications in event-based vision. Furthermore, our approach enables us to leverage cross-modal transfer learning from pre-trained frame-based models, as demonstrated in Chapter 5, enhancing model convergence and effectiveness in event-based detection tasks.

6.2.2 Advancements in Event-Based Object Detection

Event-based object detection is an emerging field that leverages the unique properties of event-based sensors. These sensors are inherently immune to motion blur and excel in low-light scenarios, presenting novel opportunities for object detection [101]. Early works in this space have focused on reconstructing intensity images from events to provide compatibility with established CV methodologies [147]. However, this method often entails significant computational demands and risks generating unrealistic reconstructions.

A shift towards directly processing event streams, foregoing the need for frame reconstruction, has gained momentum. This approach utilizes spatial and spatio-temporal encoding methods

allowing for the application of established CV tasks to event data [29]. Despite the scarcity of labeled event-based datasets, recent studies have shown promising results by adapting established CV models to event data. For instance, adaptations of architectures like YOLOv3 [149] have shown real-time performance capabilities and comparable results to traditional frame-based approaches [105, 95]. Our research extends these efforts by examining the influence of various image-like event representations on the task of object detection. We aim to evaluate their proficiency in harnessing the unique properties of event data for enhanced object detection performance.

6.2.3 Fusion Methods for Event-Based Multi-Modal Object Detection

The fusion of frame-based and event-based data is increasingly recognized as a promising method for enhancing object detection, particularly in challenging environmental conditions. Literature in this domain has introduced a variety of fusion techniques, generally leading to improved accuracy but often at the expense of increased computational demands and slower inference times.

Chen's work [32] on applying pseudo-labels for transfer learning and exploring simple fusion techniques using parallel object detectors is noteworthy. Similarly, Li *et al*. [95] proposed a joint framework combining frames and event data using convolutional spiking neural networks and YOLOv3 architecture, achieving significant improvements in nighttime object detection performance.

Advanced fusion methods at the feature level, as explored by Cao et al. [26] and Liu et al. [105], further demonstrate the potential of integrating features from both modalities to enhance detection capabilities. These works suggest that while multi-modal fusion often leads to better accuracy, it can impact inference speeds due to the added computational load.

Aligned with these advancements, our study evaluates two distinct fusion approaches—early and late fusion—across various image-like event representations. Early fusion integrates modalities at the input level, while late fusion combines decisions from separate models for each modality. This investigation aims to understand the impact of these fusion methods on object detection performance, providing insights into the potential and limitations of combining different modalities

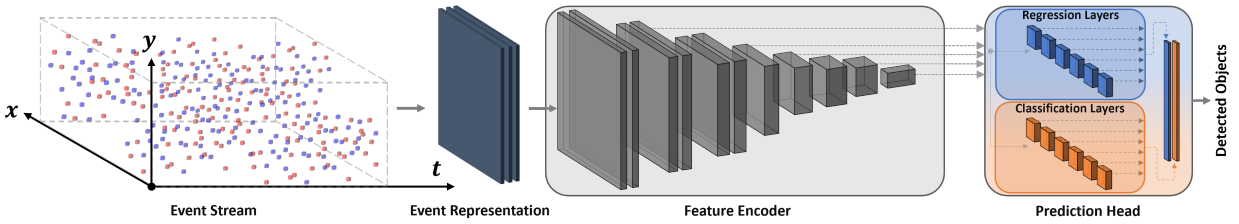


Figure 6.1: Demonstration of the event-based object detection framework utilized in this work. The asynchronous event stream is converted to a dense image-like representation using different encoding methods. The resulting 3-channel output is fed into the object detection model to identify and localize objects in the scene. The presented object detection model is based on the SSD architecture [108].

for enhanced detection capabilities.

6.3 Methodology

6.3.1 Event-Based Object Detection

Building on the foundations set in Chapter 5, this section explores the application of the Compact Spatio-Temporal Representation (CSTR) for the task of object detection in event-based vision. Our approach leverages the robust spatio-temporal encoding of the CSTR to enhance object detection algorithms' capability to interpret dynamic scenes captured by event-based sensors.

The CSTR, detailed in Section 5.3.3 of Chapter 5, effectively captures the spatial, temporal, and polarity information in a 3-channel image-like format. This representation's compatibility with standard computer vision architectures facilitates its integration into existing object detection frameworks as well. Accordingly, we compare the performance of the CSTR vs foundational image-like representations, defined in Section 5.3.2 of Chapter 5, for the task of object detection. The event-based object detection framework is presented in Figure 6.1. This comparison aims to demonstrate the efficacy of CSTR in capturing relevant features for object detection in comparison to other image-like event representations.

6.3.1.1 Integrating the CSTR for Object Detection

Incorporating image-like representations, such as the CSTR and others discussed in Chapter 5, into object detection tasks necessitates specific adaptations. The first step involves transforming the sampled event batches into their respective image-like formats. For instance, in the case of the CSTR, this transformation results in a 3-channel representation akin to a conventional RGB image. These transformed batches (or event-representation frames) are then used as inputs to the object detection models.

Object detection differs from the action recognition or classification tasks discussed previously, as it requires not only identifying objects within a frame but also accurately localizing them. Moreover, object detection tasks frequently involve scenes with multiple objects, necessitating simultaneous detection and classification of each distinct item. The spatial accuracy and contour sharpness, particularly emphasized in the CSTR, play a pivotal role in this task. This necessity holds true for other image-like representations as well, where the clarity and precision of the representation significantly influence the model's ability to detect and classify objects effectively. Therefore, the quality of the transformed event batches, in terms of spatial accuracy and temporal information retention, is critical in the context of object detection.

6.3.1.2 Object Detection Models

For our object detection experiments, we utilize one-stage object detectors, specifically the Single Shot MultiBox Detector (SSD) model [108]. SSD is chosen for its well-established balance between latency and accuracy, making it suitable for real-time object detection tasks. This model's architecture allows for effective processing of the 3-channel inputs generated by representations like the CSTR, without requiring any modifications to its architecture. The SSD's ability to efficiently handle these transformed event-based inputs is a key factor in its selection, as it aligns with the need for prompt and precise object detection in dynamic scenes captured by event-based sensors. Nevertheless, other object detectors can be used as needed..

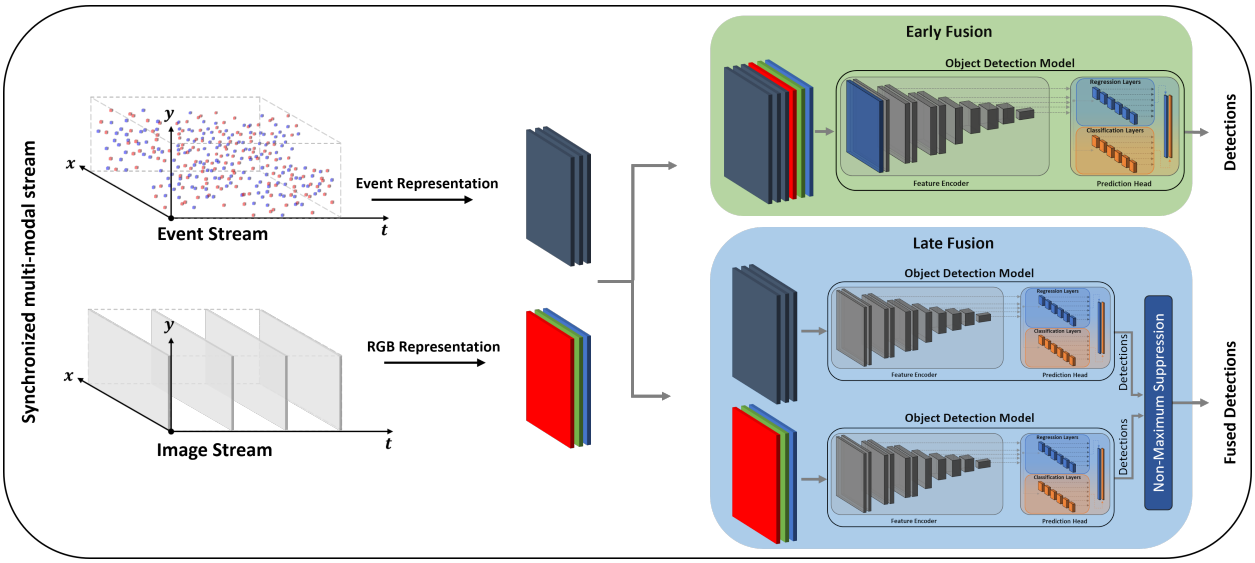


Figure 6.2: Diagram of the multi-modal framework showcasing the fusion methods applied in this work. Early fusion merges the modalities' representations at the input stage, while late fusion employs parallel detection streams, integrating their outputs through Non-Maximum Suppression.

6.3.1.3 Training and Fine-Tuning

Given the unique nature of event-based data, the pre-trained weights of these models, typically trained on standard RGB datasets [88], are fine-tuned on datasets transformed using CSTR and other image-like event representations. This process allows the models to adapt to the spatio-temporal dynamics inherent in event-based vision, potentially improving their detection capabilities in challenging environments.

6.3.2 Multi-Modal Object Detection

In this section, we explore multi-modal object detection by integrating event-based and frame-based sensor data. This multi-modal approach aims to leverage the complementary strengths of both modalities to enhance detection performance, especially in environments where either modality alone may face limitations. We focus on two primary fusion techniques: early fusion and late fusion, to examine how combining these modalities impacts object detection performance.

6.3.2.1 Early Fusion

Early fusion involves the combination of data from the different modalities at the input stage. In this fusion method, data transformed from each sensor type is merged before being input into the detection model. For example, we concatenate the CSTR-transformed event data with corresponding frame-based images, creating a unified multi-channel input. This early-stage integration enables the detection model to learn and extract features from both modalities simultaneously.

In practical terms, considering a standard 3-channel frame-based image and a 3-channel event-representation frame (such as the CSTR), we combine these along their channel dimension. This results in a composite input with a resolution of $6 \times W \times H$, as shown in Figure 6.2, effectively doubling the information that the object detection model receives. However, as these models are typically designed for a 3-channel input, adapting them to accept a 6-channel input necessitates modifications to the initial input convolutional layer. To support the 6-channel composite input, we make the necessary adjustments to the model's first layer, as detailed in Section 4. This approach aims to harness a richer set of features for the model, potentially leading to enhanced robustness and accuracy in object detection across various environmental conditions.

6.3.2.2 Late Fusion

Late fusion, also known as decision-level fusion, involves processing each modality through separate models and merging their decisions at the final stage. This technique can be advantageous when each sensor type offers unique and complementary information. For instance, frame-based cameras provide rich texture information, while event-based sensors excel in capturing HDR and motion scenes.

In our setup, as illustrated in Figure 6.2, we deploy two separate instances of the object detection model — one for the frame-based modality and another for the event-based modality. Each model is trained using its modality-specific data but with the same set of labels, aiming to minimize a combined loss that sums the individual regression and classification losses from each model.

During both validation and testing, the models' outputs -consisting of BBs, confidence scores,

Table 6.1: Comparison of model parameters for different fusion methods based on PyTorch's implementation of SSD300-VGG16[1] [108]. The base model is trained on the COCO dataset containing 91 different object classes [103].

Model Configuration	Number of Parameters
Base Model	35,641,826
Early Fusion	35,643,554 (+1,728)
Late Fusion	71,283,652 (+35,641,826)

and classification labels- are fused using NMS. NMS is a standard post-processing step in object detection frameworks that refines the detection results by discarding redundant BB based on overlap metrics and confidence scores [74, 108, 152]. Specifically, it operates by selecting the BB with the highest confidence score while removing any overlapping BBs that have an intersection-over-union higher than a predefined threshold. Throughout this chapter, we adopt a standard NMS threshold of 50%, aiming to achieve an optimal balance between precision and recall.

This training approach, which simultaneously tunes detectors for both modalities, allows for the exploration of potential cross-modal learning dynamics. Although more advanced post-processing techniques exist, our focus remains on evaluating the influence of different event-based image representations on the efficacy of multi-modal object detection methods.

In general, the choice between early and late fusion methods involves considering the computational overhead and the impact on trainable parameters. Early fusion is characterized by minimal overhead, with the primary increase in trainable parameters occurring in the first convolutional layer due to the expanded multi-channel input. All subsequent layers remain unaffected, preserving the original model complexity. This expansion is quantified in Table 6.1. In contrast, late fusion carries the highest overhead as it necessitates training two separate models for the different modalities, as also detailed in Table 6.1. This effectively doubles the number of trainable parameters, increasing both the computational resource requirements and the complexity of the training process. Therefore, the selection between early and late fusion should be informed by the specific requirements of the task and available computational resources.

[1] Available at https://pytorch.org/vision/main/models/generated/torchvision.models.detection.ssd300_vgg16

6.3.3 Augmentation Methods

Building on the event-based data augmentation framework introduced in Chapter 5, we adapt and extend these techniques for object detection tasks in this chapter. While spatial and polarity inversion augmentations are adopted as in Chapter 5, we introduce the random event-drop augmentation and modify the temporal-shift augmentation. These modifications are specifically tailored to the object detection task, where BB localization is crucial, and thus, preserving the most recent events is essential for accurate predictions. Overall, these augmentations are crucial for improving model robustness and generalization by introducing variability in the training data.

6.3.3.1 Spatial Augmentations

Spatial augmentations play a crucial role in enhancing the performance of our object detection models. These transformations, including translation, rotation, scaling, and flipping, are designed to make the model invariant to various spatial variations. Such invariance is particularly vital in event-based vision, where spatial characteristics of the data can fluctuate significantly, potentially impacting the model's accuracy and reliability.

In our work, we apply these spatial augmentations uniformly across both modalities — the generated image-like event-representation frames and the conventional frame-based images. This consistent application across different data types is critical when utilizing a multi-modal approach, ensuring that all inputs to the model undergo similar preprocessing. This not only maintains consistency in how the data is treated but also contributes to the model's ability to adapt and perform reliably across a range of spatial scenarios.

6.3.3.2 Polarity Inversion

Polarity inversion uniformly inverts the polarity of events in a given batch, effectively simulating reverse lighting conditions or motion directions. The polarities of the events in a given batch are inverted when the threshold θ_p is met (typically set as 0.5). This augmentation helps the model become robust to changes in event polarity, which can occur due to various environmental factors.

6.3.3.3 Random Event Drop

Random event drop augmentation removes a randomly selected subset of events from a given batch. The percentage of events dropped dynamically varies between 0 and a predefined maximum threshold θ_d, mimicking environments with sparse visual cues.

6.3.3.4 Temporal-Drop Augmentation

The temporal-drop augmentation, a modification of the temporal-shift method introduced in Chapter 5, shifts and drops events within a batch to simulate different temporal history dynamics. Crucially, this method only shifts events backward in time, retaining the most recent events which are vital for accurately localizing objects at their latest positions. Thus, the batch-sampling period ΔT is effectively reduced as follows:

$$\Delta T' = \Delta T + \theta_t(\lambda \Delta T), \tag{6.1}$$

where λ denotes the proportion of the temporal shift that is randomly specified during training (in the range of $[-1, 0]$) and θ_t is the max temporal shift threshold ($\theta_t \in (0, 1)$). Similarly, each event's timestamp is modified as shown in Equation 5.11, Chapter 5. Then, the events with a negative timestamp (*i.e.*, $\notin [0, \Delta T']$) are removed. This technique, visualized in Figure 6.3, is crucial for learning features invariant to dynamic temporal durations and sampling times without affecting the labels' BB integrity. Further, it is applied before the transformation of events into their respective image-like representations.

Furthermore, in this chapter, we optimize the application of these augmentation methods. Instead of a fixed application, each method is randomly selected per sample based on predefined probabilities, enhancing the variability during training. For instance, spatial augmentations can have a higher likelihood of being applied (e.g., 70% of the time) compared to other augmentation methods (e.g., set as 50%). Additionally, each sample in a training batch is augmented with a different set of augmentations, each with varying randomized magnitudes, rather than applying the

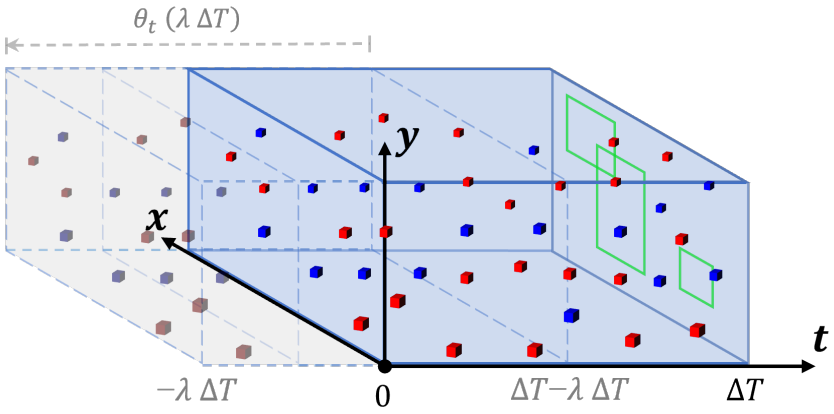


Figure 6.3: Illustration of the Temporal-Drop augmentation method, demonstrating how events are temporally shifted and then dropped, effectively simulating different temporal history dynamics. The green rectangles at normalized time $t = \Delta T$ represent the true objects' bounding boxes annotated at the end of the sample time. The maximum temporal shift θ_t demonstrated here is set as 0.5.

same settings to the entire training batch of inputs. This randomization ensures that each sample within a training batch is augmented differently, increasing the diversity of the training data and aiding in the model's ability to generalize to a wider range of scenarios.

Overall, these augmentation methods provide a comprehensive approach to introduce variability and robustness in our event-based and multi-modal object detection frameworks, significantly contributing to the effectiveness of the models in diverse and challenging environments.

6.4 Experiment Settings

This section details the datasets used, the object detectors employed, the hyperparameters selected, and the evaluation metrics utilized for the experiments.

6.4.1 Datasets Utilized

In our experiments, we employ two primary multi-modal object detection datasets: MEVDT and PKU-DDD17-CAR [95]. Both manually labeled datasets are detailed in Table 6.2 and provide unique environments and challenges for object detection in the automotive domain.

Table 6.2: Key statistics and characteristics of the multi-modal event-based object detection datasets used in our experiments. The presented datasets are captured in real-world scenarios and are manually labeled.

Parameter	**Dataset**	
	MEVDT	PKU-DDD17-CAR [95]
Number of classes	1	1
Dataset Type	Static (Traffic Sequences)	Dynamic (in-vehicle camera)
Modalities	Events & Grayscale Images	Events & Grayscale Images
Event Camera	DAVIS 240 [21]	DAVIS 346
Frame Dimension ($W \times H$)	240×180	346×260
Labeling Frequency (Hz)	24	∼1
Total Sequences	63	14
Total Labeled Samples	12759	3154
# Training Samples	10132 (79%)	2241 (71%)
# Testing Samples	2627 (21%)	913 (29%)
Total Objects	9891	5760
Total Duration (s) *	545.0	63.0
Average # events/sample †	428.4	4796.6
Batch sampling duration (ms)	43, 100, 200, 500 ‡	20

† Calculated using the longest sample size possible without overlap between adjacent samples.
‡ Samples are manually extracted from extended sequences of event data based on the specified sampling durations.
* Total duration refers to the cumulative length of all event sequences, representing the total time covered by the event data in each dataset.

MEVDT, introduced in Chapter 2, is a multi-modal dataset focusing on vehicle detection and tracking. It comprises 63 sequences recorded using a DAVIS-240c camera [21], providing synchronized image and event data. The dataset is split into training (51 sequences) and testing (12 sequences) splits, ensuring balanced representation across different parameters. The dataset, manually annotated at 24 Hz, includes static vehicles (*i.e.*, parked vehicles) that are not labeled. Our models are trained to disregard these stationary objects, instead of cropping the images during training, to minimize the distortions resulting from resizing the frames.

PKU-DDD17-CAR [95] is derived from the larger DDD17 dataset [17], which contains extensive driving data recorded across European roads using a DAVIS-346B camera. This subset consists of 14 sequences (7 for training, 7 for testing), and provides 3154 manually labeled samples of synchronized grayscale images and events. These labels were added to the originally unlabeled

DDD17 [17] dataset for object detection applications. The sequences are categorized based on lighting conditions, identified as either day-time or low-light conditions. Within these, 5 of the 14 sequences (comprising 2 in the training set and 3 in the testing set) are classified under low-light conditions, generally indicative of evening or early night hours. The remaining sequences are categorized as daytime. The overall distribution of the total samples across the training and testing sets is approximately 78% for daytime and 22% for low-light conditions.

Comparing these datasets, MEVDT offers a higher total dataset duration and more labeled samples but features fewer events per sample due to the static camera setup. In contrast, PKU-DDD17-CAR [95], recorded from a moving vehicle, presents a higher event density per sample. The distinct characteristics of these datasets provide a comprehensive testing ground for our object detection methods, particularly when exploring the effects of different batch-sampling durations on performance. In the case of MEVDT, we extract samples by selecting the last ΔT events prior to each label timestamp t_s, with ΔT varying between 43 ms and 500 ms. Accordingly, each sample contains events within the interval $[t_s - \Delta T, t_s]$. We adapt to longer sampling durations by excluding samples with insufficient temporal history (where the sample's duration is $< \Delta T$), resulting in fewer samples for longer durations. However, this method of varying ΔT could not be applied to the PKU-DDD17-CAR dataset, as its samples are pre-extracted with a fixed sampling period of approximately 20 ms. This limitation arises because PKU-DDD17-CAR does not include longer sequences of event data, restricting our ability to experiment with different sampling durations for this dataset.

Table 6.2 provides an overview of the datasets' key statistics, highlighting the differences in data types, sensor specifications, sample dimensions, and annotation methods, among other parameters. These variations in dataset characteristics are valuable in testing the robustness and adaptability of our object detection methodologies under varying conditions.

6.4.2 Representations

As in Chapter 5, we adopt the image-like event representations detailed in Section 5.3, including the CSTR developed in that chapter. These representations can be categorized as *spatial* (Event Frame and Event Count variants) or *spatio-temporal* (Timestamp Image and CSTR variants). The objective of this chapter is to assess the impact of these different representations on object detection, a critical task in CV. This evaluation extends to both multi-modal scenarios and baseline image-only setups.

Given our earlier findings (Chapter 5), we restrict our experiments to 3-channel formats of each image-like event representation. This decision aligns with our goal to maintain compatibility with standard CV architectures and provide an optimal use of pre-trained weights. For the grayscale images captured by the APS, we utilize a 3-channel RGB format, where the intensity information is duplicated across all three channels, ensuring uniformity in input data format.

6.4.3 Object Detectors

In our experiments, we employ the SSD [108] with a VGG16 encoder [160] as our base object detection model. This choice is motivated by SSD's ability to balance accuracy and processing latency, making it well-suited for real-time detection tasks. We base our model on Torchvision's implementation of SSD300 with a VGG16 encoder[2], pre-trained on the COCO dataset [103]. This pre-training is essential to ensure better model convergence and generalization, particularly advantageous for the event-based modality where the availability of extensive labeled training data is limited.

Given that the COCO dataset encompasses 91 object classes [103], we modify the SSD's classification head to be compatible with the number of classes in our datasets, while keeping the regression head unchanged to preserve as much of the pre-trained weights as possible. This approach maintains a strong foundation for the model, leveraging the generalization capabilities provided by the COCO dataset.

[2]https://pytorch.org/vision/main/models/generated/torchvision.models.detection.ssd300_vgg16

For the multi-modal experimental setups, both early and late fusion methods require further adaptations to the modified base model to accommodate the different data formats and fusion strategies.

In the early fusion experiments, we adapt the first convolutional layer of the SSD [108] to process 6-channel inputs instead of the standard 3-channel inputs. To retain the benefits of the pre-trained weights, we replicate them across the additional channels in the first layer, ensuring a seamless transition to the expanded input format. This modification is crucial to accommodate the concatenated event and frame-based input data without compromising the initial training advantages.

In late fusion, our setup employs two instances of the modified SSD model, each responsible for one modality. These models are trained simultaneously, optimizing a combined loss function that aggregates the individual losses from each model through simple addition. This dual-model approach allows us to explore the potential of late fusion in enhancing object detection performance, especially in scenarios where each sensor modality offers distinct advantages.

In all cases, the event-representation frames and grayscale images are uniformly resized to a resolution of 300×300, to accommodate the requirements of the object detection model. This step ensures that the input data is consistent in dimensions across different representations and modalities.

6.4.4 Training Hyperparameters

For our experiments, we carefully select specific hyperparameters to ensure consistency, replicability, and optimal model performance. These settings are detailed as follows:

- **Batch Size:** A fixed batch size of 32 is maintained throughout the training process.
- **Optimizer:** The ADAM optimizer [87] is chosen, with key parameters set as:
  - *Learning Rate:* An initial learning rate of 1×10^{-4} is used.
  - *Weight Decay:* A weight decay parameter is set at 1×10^{-5}.

- **Normalization:** ImageNet [88] normalization parameters (mean and standard deviation) are applied, consistent with the approach in Chapter 5.
- **Training Duration:** Models are trained for up to 50 epochs. Early stopping is employed, triggered after 10 consecutive epochs without improvement in validation performance.
- **Model Selection:** The best model is determined based on the highest mAP on the validation set during the training phase.
- **Dataset Splits:** We utilize the official training and testing splits as provided with each dataset (refer to Chapter 2 and Ref. [95]).
- **Validation Set Selection:**
  - For MEVDT, we employ a sequence-based split to minimize data leakage and improve generalization performance. The sequences are sorted by sample count and then assigned to the training set until reaching approximately 80% of the total samples. The remaining sequences, containing around 20% of the samples, are assigned to the validation set. This approach is followed for all sampling period configurations used in our experiments.
  - For PKU-DDD17-CAR [95], the validation set is selected by randomly allocating 80% of the samples to the training set and 20% to the validation set, considering the high variance of sequence sizes and the sequentially inconsistent nature of the dataset's samples. Additionally, in the final evaluation, we assess performance across all test sequences and provide separate results for both day-time and low-light conditions using the same evaluation metrics. This approach allows for a comprehensive evaluation of the model's performance under varying lighting conditions, alongside the overall performance.
- **Data Shuffling:** Shuffling of the training set is done after each epoch to prevent the model from learning false sequential patterns.

- **Augmentations:** Augmentations are a critical component of our training process, with configurations varied for each training instance:

  - *Spatial Augmentation:* Selected with a 70% probability, including randomized crops (70%–100% of the original size) with aspect ratios between 3/4 to 4/3. When spatial augmentations are applied, horizontal flips are included with a 50% probability. Rotations are excluded to avoid adverse effects on performance.
  - *Temporal-Drop Augmentation:* Implemented with a 50% probability, with a maximum temporal shift threshold (θ_t) set at 0.5, affecting up to 50% of the temporal range.
  - *Polarity-Inversion Augmentation:* Applied with a 50% probability, where the polarity inversion threshold (θ_p) is set to 0.5.
  - *Event Drop Augmentation:* Implemented with a 50% probability, with a maximum event drop percentage set to 30% (θ_d set at 0.3).

These hyperparameters are chosen to optimize the models' performance and generalizability across different datasets and object detection scenarios presented in our experiments.

6.4.5 Evaluation Metrics

Mean Average Precision (mAP) is a widely adopted benchmark metric for object detection models [103, 49]. Contrary to image classification tasks that assign a single label to an image, object detection challenges involve recognizing, categorizing, and precisely locating multiple objects within an image, each defined by a unique BB. Due to the multitude of potential True Negative (TN) classifications, conventional metrics like accuracy are unsuitable for object detection.

IoU, also known as the Jaccard Index [79], measures the accuracy of object localization. It quantifies the overlap between predicted and ground truth BBs, as depicted in Figure 6.4. The IoU is calculated by finding the ratio of the intersection area to the union area, yielding a value between 0.0 (no overlap) and 1.0 (perfect overlap).

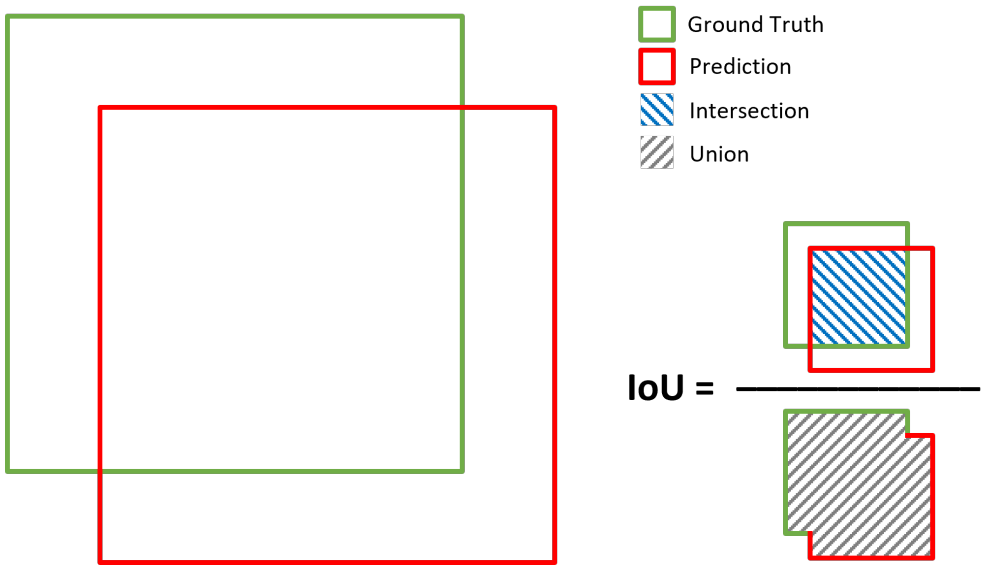


Figure 6.4: Visualization of the Intersection over Union (IoU) metric. IoU is calculated by finding the ratio of the intersection between the predicted and ground truth BBs to their union.

Predictions are classified as True Positive (TP), False Positive (FP), or False Negative (FN) based on a specified IoU threshold, α. A prediction qualifies as a TP if IoU is $\geq \alpha$, or as an FP if IoU is $< \alpha$. A true object without a matching BB exceeding the IoU threshold α is marked as a FN, while any prediction without a corresponding true object is a FP. With these prediction classifications, we are able to calculate the precision and recall using the following equations:

$$Precision = \frac{TP}{TP + FP} \tag{6.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{6.3}$$

The precision metric assesses the model's ability to correctly classify true samples as positive, focusing on the accuracy of positive classifications. On the other hand, recall measures the model's capacity to detect all true samples while ignoring instances of FP classifications.

In the context of object detection, the predictions for each object category are sorted by confidence values in descending order across all images within the evaluation set. This sorting process results in the accumulation of TPs and FPs, enabling the computation of precision and recall at

each point along this sorted list. Subsequently, the precision and recall values obtained for each class contribute to the construction of the precision-recall curve. This curve serves as the basis for calculating the AP for each individual class by measuring the area under the curve. The mAP is derived by averaging these AP values across all classes.

The mAP was originally computed using an IoU threshold of $\alpha = 0.5$ in the PASCAL VOC challenge [49], denoted as mAP_{50}. The COCO dataset extends this evaluation by considering mAP across multiple IoU thresholds, ranging from 0.5 to 0.95 in 0.05 increments [103]. The resulting range of mAP results is averaged to produce the $\text{mAP}_{[0.5:0.95]}$, which is referred to as mAP in this work. Additionally, the COCO approach utilizes a 101-point interpolation method, unlike PASCAL VOC's 11-point method, providing a more refined evaluation of mAP. In this work, we employ the mAP, mAP_{50}, and mAP_{75} metrics based on the COCO API[3] [103] for all evaluations and mAP metric computations. For readers seeking a deeper understanding of the mAP metric, a comprehensive explanation is available at [157].

6.5 Experiment Results

Following the experimental setup detailed earlier, this section presents both quantitative and qualitative analyses of the conducted experiments. Our focus here is to present a comprehensive evaluation of the performance of our proposed methods under various settings and conditions.

6.5.1 Event-based Object Detection

In this section, we present the results of our event-based object detection experiments. These findings are critical for understanding the effectiveness of various event representations in object detection. For each dataset, we demonstrate the test set results with and without the implementation of the augmentation framework.

Consistent with our findings in Chapter 5, representations that solely encode spatial informa-

[3] Available at https://github.com/cocodataset/cocoapi/tree/master

Table 6.3: Evaluation results of different event representations on MEVDT's test set across multiple batch-sampling durations. The best result per column is in highlighted **bold**.

Representation	mAP					mAP_{75} (%)					mAP_{50} (%)				
	43 ms	100 ms	200 ms	500 ms	**avg.**	43 ms	100 ms	200 ms	500 ms	**avg.**	43 ms	100 ms	200 ms	500 ms	**avg.**
Binary Event Frame	74.8	77.3	74.3	72.7	74.8	88.6	91.2	88.8	81.6	87.5	93.6	96.3	95.9	95.0	95.2
Polarized Event Frame	77.7	78.3	79.7	80.1	79.0	90.8	**92.8**	92.1	93.7	92.4	94.6	96.4	96.5	96.7	96.0
Binary Event Count	75.9	76.8	75.5	73.6	75.4	89.5	92.1	86.4	81.3	87.3	93.2	96.0	95.6	95.9	95.2
Polarized Event Count	77.3	78.6	78.0	76.1	77.5	90.1	91.9	92.0	86.1	90.0	94.7	96.4	96.4	96.7	96.0
Timestamp Image	75.2	79.0	**81.9**	80.8	79.2	88.3	92.6	**94.3**	**94.5**	92.4	94.7	**96.5**	97.1	97.5	96.4
Timestamp Image & Count	77.6	**79.9**	80.5	80.6	79.7	91.1	92.2	93.0	93.2	92.4	94.6	96.0	97.0	97.0	96.1
CSTR (mean $\bar{T}_s$ only)	78.7	79.2	81.3	**81.5**	**80.2**	90.9	91.7	**94.3**	**94.5**	92.8	95.4	96.2	**97.3**	**97.7**	**96.6**
CSTR (mean $\bar{T}_s$ & Count)	**78.9**	79.3	79.8	80.5	79.6	**92.0**	92.2	93.1	**94.5**	**93.0**	**95.7**	95.9	96.3	97.5	96.3

Table 6.4: Evaluation results of different event representations in addition to the proposed augmentation framework on MEVDT's test set across multiple batch-sampling durations.

Representation	mAP					mAP_{75} (%)					mAP_{50} (%)				
	43 ms	100 ms	200 ms	500 ms	**avg.**	43 ms	100 ms	200 ms	500 ms	**avg.**	43 ms	100 ms	200 ms	500 ms	**avg.**
Binary Event Frame	80.1	81.6	78.8	76.2	79.2	90.6	92.2	89.9	86.1	89.7	95.4	97.0	97.3	97.1	96.7
Polarized Event Frame	81.7	82.7	82.7	82.7	82.5	**92.0**	92.5	**93.8**	**94.8**	**93.3**	96.0	**97.9**	**98.4**	98.7	**97.8**
Binary Event Count	78.5	79.6	77.7	75.6	77.8	89.5	92.4	90.8	82.4	88.8	94.5	97.2	97.7	96.3	96.4
Polarized Event Count	81.0	82.4	79.3	77.1	79.9	90.7	92.8	90.6	86.3	90.1	96.2	97.7	97.3	97.3	97.1
Timestamp Image	**81.8**	81.4	83.8	83.2	82.6	91.5	92.6	93.7	94.7	93.1	96.2	97.4	**98.4**	97.5	97.4
Timestamp Image & Count	79.6	**83.7**	83.7	84.6	82.9	91.7	**93.4**	93.5	93.9	93.1	95.9	97.4	98.2	98.7	97.5
CSTR (mean $\bar{T}_s$ only)	79.1	82.2	84.1	84.2	82.4	90.7	92.7	93.6	93.4	92.6	95.7	97.3	98.3	98.7	97.5
CSTR (mean $\bar{T}_s$ & Count)	80.4	82.7	**84.9**	**84.9**	**83.2**	91.6	92.5	93.5	94.5	93.0	**96.5**	96.5	98.1	**98.8**	97.5

The best result per sampling duration and metric is in highlighted **bold**.

tion, such as Event Frame representations, tend to underperform with extended sampling durations, as shown in Table 6.3. For example, the Binary Event Frame representation consistently shows reduced effectiveness under prolonged durations, particularly with precision-sensitive metrics like mAP and mAP_{75}. This trend is also noticeable in the Binary Event Count representation, underscoring the significance of incorporating event polarity to minimize signal saturation. Conversely, the Polarized Event Count and Event Frame representations, while not deteriorating significantly, do not exhibit robustness or marked improvement with longer batch-sampling durations. In contrast, spatio-temporal representations, primarily the Timestamp Image and CSTR variants, significantly benefit from extended durations. This observation highlights their resilience to varying batch periods and their superior capability in encoding the rich temporal information of event data, compared to other image-like representations. The CSTR, in particular, demonstrates robust performance across all metrics, with both of its variants achieving the best average performance in nearly all metrics.

Interestingly, including the event count channel in both the Timestamp Image and the CSTR

did not consistently result in better performance. This contrasts with our earlier results in Chapter 5, where these representations significantly benefited from the added context provided by the event count. We hypothesize that this discrepancy arises from the differing nature of the tasks, specifically between image classification and object detection. Object detection necessitates a clear signal for BB regression and classification, whereas image classification involves categorizing the input as a whole. Consequently, DL-based classification networks can extract useful global context from this additional information, while object detectors might require more nuanced low-level input feature extraction methods.

When integrating the augmentation framework, as demonstrated in Table 6.4, spatio-temporal representations consistently outperform others, even with the inclusion of randomized augmentations. They also consistently benefit from longer batch-sampling periods across all metrics. The CSTR, in particular, exhibits exceptional performance, yielding the best average results under the mAP metric. Interstingly, the Polarized Event Frame representation demonstrates strong performance under the mAP75 and mAP50 metrics, surpassing the spatio-temporal representations. This can be attributed to the static nature of this dataset, which reduces representation saturation and enables object detectors to more effectively identify objects, particularly with the addition of randomized augmentations.

The results from the PKU-DDD17-CAR dataset, as depicted in Table 6.5, present a contrasting narrative. In this evaluation, the Polarized Event Count method surpasses other event representation techniques across all metrics. This variance underscores the critical role of having ample data for training and fine-tuning event-based models. Although this dataset offers a broader range of conditions compared to MEVDT, it is limited by a significantly smaller number of samples. Consequently, event-based models encounter a scarcity of data points necessary for learning the intricate spatio-temporal patterns inherent in such representations. In contrast, a spatial representation with reduced complexity, like the Polarized Event Count, can more effectively capitalize on the limited available samples. This situation highlights the challenges posed by dataset limitations in event-based modalities and underlines the need for incorporating randomized data augmentation

Table 6.5: Evaluation results of different event representations on PKU-DDD17-CAR's test set under day-time and low-light conditions. The best result per column is in highlighted **bold**.

Representation	All Sequences			Day-Time Sequences			Low-Light Sequences		
	mAP (%)	mAP_{75} (%)	mAP_{50} (%)	mAP (%)	mAP_{75} (%)	mAP_{50} (%)	mAP (%)	mAP_{75} (%)	mAP_{50} (%)
Binary Event Frame	21.2	19.2	43.3	20.1	18.6	39.8	26.2	21.1	58.9
Polarized Event Frame	21.0	17.1	43.9	19.6	16.9	40.7	27.2	17.8	58.4
Binary Event Count	21.0	17.9	43.0	20.2	17.8	40.1	25.3	19.3	56.7
Polarized Event Count	**22.7**	**20.8**	**46.9**	**21.7**	**20.5**	**43.7**	**27.6**	**22.3**	**61.5**
Timestamp Image	21.7	19.3	44.3	20.6	19.2	41.3	27.0	20.8	58.5
Timestamp Image & Count	21.6	18.2	46.3	20.7	18.2	43.1	25.8	18.1	60.9
CSTR (mean $\bar{T}_s$ only)	21.4	18.1	44.5	19.8	17.0	41.9	25.9	17.4	57.6
CSTR (mean $\bar{T}_s$ & Count)	22.0	17.9	45.7	20.8	17.8	42.5	27.2	18.3	60.7

Table 6.6: Evaluation results of different event representations in addition to the proposed augmentation framework on PKU-DDD17-CAR's test set under day-time and low-light conditions.

Representation	All Sequences			Day-Time Sequences			Low-Light Sequences		
	mAP (%)	mAP_{75} (%)	mAP_{50} (%)	mAP (%)	mAP_{75} (%)	mAP_{50} (%)	mAP (%)	mAP_{75} (%)	mAP_{50} (%)
Binary Event Frame	24.2	21.5	49.4	23.6	22.8	46.5	28.5	19.8	63.0
Polarized Event Frame	**26.1**	22.1	**53.1**	24.9	21.6	**50.0**	**32.0**	**25.4**	**67.7**
Binary Event Count	23.9	19.6	50.4	23.3	20.0	48.1	27.9	20.0	61.7
Polarized Event Count	**26.1**	**23.8**	52.4	**25.4**	**24.3**	49.7	30.4	24.0	65.0
Timestamp Image	23.1	20.3	49.4	22.3	20.9	46.7	27.9	20.0	62.2
Timestamp Image & Count	24.7	22.7	51.1	23.7	22.7	48.3	30.5	24.6	65.1
CSTR (mean $\bar{T}_s$ only)	23.7	20.4	48.7	22.9	21.1	46.1	28.5	19.6	61.2
CSTR (mean $\bar{T}_s$ & Count)	24.2	21.9	49.9	23.3	21.8	46.9	29.2	24.3	63.8

The best result per column is in highlighted **bold**.

techniques to enhance the diversity of training data. Furthermore, we can infer that a brief sampling duration (20 ms in this case) yields inadequate information for the object detection model to effectively utilize spatio-temporal representations. Additionally, careful examination of the dataset labels reveals significant labeling issues, including numerous instances of poor BB alignment and frequent occurrences of unlabeled objects. These labeling inaccuracies, coupled with the dataset's limited size, significantly impede the generalization capabilities of the detection models.

Similar trends are observed when implementing augmentations, as illustrated in Table 6.6. The integration of our proposed augmentation framework significantly enhances the performance of event-based models on the PKU-DDD17-CAR dataset. Notably, the Polarized Event Count continues to excel, surpassing other methods under the mAP and mAP_{75} metrics. Intriguingly, spatial representations, encompassing both event frame and event count, demonstrate superior performance on average compared to their spatio-temporal counterparts. This observation further underscores the previously discussed limitations inherent to this dataset.

A particularly noteworthy finding with this dataset is the enhanced performance of all trained event-based models in low-light conditions. Although this outcome may be influenced by various factors, including the dataset's imbalance, it notably underscores the efficacy of event cameras in low-light scenarios. To fully harness and validate this potential in object detection applications, the development and utilization of more comprehensive and diverse datasets are crucial.

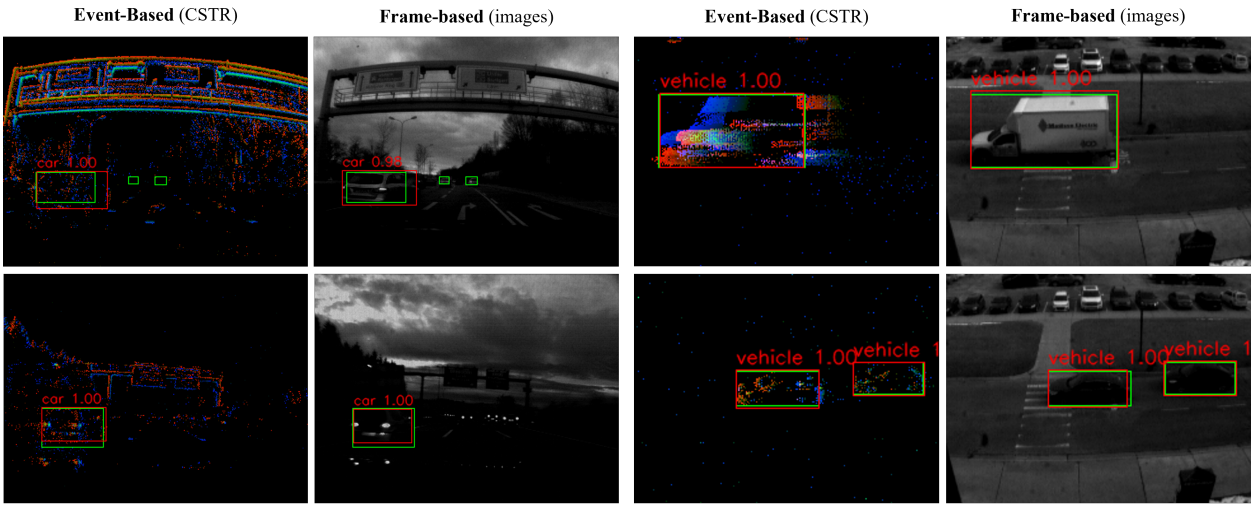
We visualize some results of our event-based approach in Figure 6.5. Here, we present the outputs of both the event-based and the frame-based baseline models using selected samples from each dataset. These results are derived from the fine-tuned SSD model, employing the CSTR for the event-based modality and intensity images for the frame-based model. Notably, in the PKU-DDD17-CAR samples shown in (a), there are evident labeling issues, including poor alignment and unlabeled objects. Meanwhile, the MEVDT dataset samples, particularly in Scene A's sequences, demonstrate the models' fine-tuning to disregard parked vehicles at the top section where no events are generated. When there is movement, the event-based approach, especially with spatio-temporal image-like representations, shows its potential in object detection by generating precise BBs, as evidenced in comparison to the ground truth labels.

6.5.2 Multi-Modal Object Detection

In this section, we compare the results of the multi-modal fusion methods described in Section 6.3. Additionally, we compare these results with a baseline frame-based SSD model, fine-tuned using each dataset's respective images. We also explore the impact of our proposed augmentation framework on these fusion methods in a multi-modal context. It is important to note that when applying augmentations, only spatial augmentations are applied to the frame-based modality, using the same randomly generated parameters as those applied to the event-representation frames.

6.5.2.1 Early Fusion

Our initial evaluation focuses on early fusion methods. Using the MEVDT dataset, as shown in Table 6.7, we observe that the baseline frame-based approach generally surpasses the early fusion-



(a) PKU-DDD17-CAR (b) MEVDT ($\Delta T = 500\ ms$)

Figure 6.5: Visualization of selected object detection results obtained by fine-tuning a pre-trained SSD model [108], in combination with our proposed augmentation framework, for both event-based and frame-based modalities. The results are showcased using the datasets featured in this chapter, specifically, (a) PKU-DDD17-CAR [95], and (b) MEVDT (presented in Chapter 2) sampled at 500 ms. The event-based approach employs the CSTR as the chosen event representation, while the frame-based approach uses intensity images. Ground truth and predicted objects are depicted in green and red, respectively.

based multi-modal approach. This discrepancy can be attributed to several factors, including the nature of the dataset. Since the camera in this dataset is stationary, most events captured are directly related to the movement of objects (*i.e.*, vehicles) within the scene. Given the varying speeds of the vehicles, specifically the instances where they come to a complete stop, the event-based sensor can provide an unreliable signal in these scenarios. Consequently, this dataset may not offer adequate data for the model to learn to identify these patterns and prioritize modalities, even at longer sampling durations. The fusion occurring at the input level, combined with the lack of intricate fine-tuning methods, may prevent the initial convolutional layer from learning these patterns effectively. As a result, the early-fusion models may converge to a less optimal global minimum compared to the frame-based models, which find better convergence points more readily.

Table 6.8 presents the results of incorporating our augmentation framework. Interestingly, the proposed augmentations significantly enhance the performance of basic spatial event representa-

Table 6.7: Evaluation results of the early-fusion multi-modal approach on MEVDT's test set. The results are demonstrated using different event representations across multiple batch-sampling durations.

Representation	**mAP**					**mAP$_{75}$ (%)**					**mAP$_{50}$ (%)**				
	43 ms	100 ms	200 ms	500 ms	**avg.**	43 ms	100 ms	200 ms	500 ms	**avg.**	43 ms	100 ms	200 ms	500 ms	**avg.**
Binary Event Frame	84.3	84.7	85.8	83.8	84.6	96.0	95.9	96.7	95.8	96.1	98.9	98.9	98.9	98.9	**98.9**
Polarized Event Frame	84.1	84.2	83.0	83.1	83.6	96.5	95.7	95.5	96.3	96.0	98.5	98.7	97.8	98.3	98.3
Binary Event Count	84.9	82.6	80.8	78.2	81.6	95.9	95.7	93.5	87.8	93.3	**99.0**	98.9	97.9	97.0	98.2
Polarized Event Count	84.9	83.7	85.5	82.0	84.0	95.6	95.6	95.9	93.3	95.1	98.7	98.7	97.9	97.0	98.1
Timestamp Image	85.9	87.2	84.1	86.8	86.0	95.7	96.8	95.8	96.9	96.3	98.7	98.9	98.8	98.9	98.8
Timestamp Image & Count	84.6	85.7	85.9	86.6	85.7	96.8	96.9	96.9	97.0	96.9	98.9	98.9	98.9	**99.0**	**98.9**
CSTR (mean $\bar{T}_s$ only)	84.7	86.6	84.2	87.7	85.8	96.4	**97.0**	96.8	97.0	96.8	98.4	98.9	98.8	**99.0**	98.8
CSTR (mean $\bar{T}_s$ & Count)	86.0	86.3	84.5	86.5	85.8	95.8	96.6	96.9	97.0	96.6	98.9	**99.0**	98.9	**99.0**	**98.9**
Baseline (images only)	**88.2**	**88.5**	**86.6**	**88.4**	**87.9**	**97.9**	96.8	**97.9**	**97.8**	**97.6**	98.9	98.9	**99.0**	98.8	**98.9**

The best result per sampling duration and metric is in highlighted **bold**.

Table 6.8: Evaluation results of the early-fusion multi-modal method in addition to the proposed augmentation framework on MEVDT's test set. The results are demonstrated using different event representations across multiple batch-sampling durations.

Representation	**mAP**					**mAP$_{75}$ (%)**					**mAP$_{50}$ (%)**				
	43 ms	100 ms	200 ms	500 ms	**avg.**	43 ms	100 ms	200 ms	500 ms	**avg.**	43 ms	100 ms	200 ms	500 ms	**avg.**
Binary Event Frame	87.8	87.8	88.5	**89.5**	**88.4**	95.8	95.8	**96.9**	97.0	96.4	**98.9**	98.8	**99.0**	**99.0**	**98.9**
Polarized Event Frame	89.1	**88.3**	88.1	86.2	87.9	**96.9**	**96.9**	95.9	96.9	96.7	**98.9**	**98.9**	98.9	98.9	**98.9**
Binary Event Count	86.1	84.1	87.0	86.9	86.0	**96.9**	95.8	95.9	95.9	96.1	**98.9**	98.7	98.8	**99.0**	98.8
Polarized Event Count	88.1	87.4	87.5	88.2	87.8	96.7	**96.9**	95.9	96.7	96.6	98.7	**98.9**	98.8	98.9	98.8
Timestamp Image	88.2	87.3	87.9	88.0	87.9	96.7	96.6	96.8	96.9	96.7	98.7	98.6	98.8	98.9	98.7
Timestamp Image & Count	88.0	87.1	86.2	87.1	87.1	96.8	96.7	96.0	97.0	96.6	98.8	98.7	98.9	98.9	98.8
CSTR (mean $\bar{T}_s$ only)	87.9	86.4	88.4	89.4	88.0	96.7	96.7	95.9	96.9	96.5	98.7	98.7	98.8	98.8	98.8
CSTR (mean $\bar{T}_s$ & Count)	88.9	87.4	87.0	87.1	87.6	**96.9**	**96.9**	95.7	**97.9**	**96.9**	**98.9**	**98.9**	98.7	**99.0**	**98.9**
Baseline (images only)	**89.3**	88.2	**89.7**	86.4	**88.4**	96.4	96.7	96.7	**97.9**	**96.9**	98.4	98.7	98.7	98.9	98.7

The best result per sampling duration and metric is in highlighted **bold**.

tions, more so than the spatio-temporal ones. This leads to comparable average performance across all event representation choices, particularly under the mAP_{75} and mAP_{50} metrics. In this dataset, the Binary Event Frame shows potential in complementing the intensity images of the frame-based modality. Although these multi-modal solutions underperform compared to the frame-based baseline, this can be partially attributed to the detection model's limitations in effectively leveraging larger inputs, necessitating more advanced fusion methods and capable detection models. Additionally, the nature of the dataset suggests that intensity images alone may suffice. The spatial augmentations slightly adversely affect the baseline frame-based model, reflecting the optimal use of pre-trained weights in leveraging transfer learning for object detection models on this dataset. The MEVDT dataset's relatively simple scenes for frame-based object detection indicate that performance might have reached its peak.

Table 6.9: Evaluation results of the proposed early-fusion multi-modal method on PKU-DDD17-CAR's test set. The results are also demonstrated under daytime and low-light conditions.

Representation	All Sequences			Day-Time Sequences			Low-Light Sequences		
	mAP (%)	mAP_{75} (%)	mAP_{50} (%)	mAP (%)	mAP_{75} (%)	mAP_{50} (%)	mAP (%)	mAP_{75} (%)	mAP_{50} (%)
Binary Event Frame	41.8	37.7	84.5	42.5	38.7	85.1	40.0	34.0	83.1
Polarized Event Frame	40.3	34.6	82.2	41.0	35.8	82.6	38.2	29.6	80.9
Binary Event Count	43.1	40.3	84.9	43.8	42.1	85.3	41.4	33.3	84.3
Polarized Event Count	42.9	37.2	85.2	43.9	38.7	86.5	40.0	30.8	83.3
Timestamp Image	**45.2**	41.9	87.3	**45.9**	43.4	88.0	43.1	37.5	85.0
Timestamp Image & Count	44.6	**42.2**	87.3	45.3	**43.7**	88.2	42.5	38.7	83.0
CSTR (mean $\bar{T}_s$ only)	44.5	39.5	87.4	45.5	41.6	88.3	41.5	32.2	84.4
CSTR (mean $\bar{T}_s$ & Count)	44.4	37.7	**87.9**	45.2	39.5	**88.4**	42.1	32.5	85.7
Baseline (images only)	44.5	41.0	87.1	44.9	41.7	87.5	**43.5**	**39.1**	**86.1**

The best result per column is highlighted in **bold**.

Table 6.10: Evaluation results of the early-fusion multi-modal method in addition to the proposed augmentation framework on PKU-DDD17-CAR's test set. The results are also demonstrated under daytime and low-light conditions.

Representation	All Sequences			Day-Time Sequences			Low-Light Sequences		
	mAP (%)	mAP_{75} (%)	mAP_{50} (%)	mAP (%)	mAP_{75} (%)	mAP_{50} (%)	mAP (%)	mAP_{75} (%)	mAP_{50} (%)
Binary Event Frame	43.8	37.5	88.6	43.9	37.5	89.0	44.3	38.3	88.9
Polarized Event Frame	41.8	35.4	86.3	42.6	36.4	87.0	39.8	32.6	85.0
Binary Event Count	45.0	40.2	88.5	46.3	42.0	90.0	41.2	35.3	84.1
Polarized Event Count	46.3	41.5	89.6	46.8	42.8	90.6	44.3	37.8	86.5
Timestamp Image	45.5	41.0	89.0	46.4	42.4	90.2	42.9	37.2	86.2
Timestamp Image & Count	45.4	39.8	89.5	46.4	41.5	90.6	42.8	34.6	86.6
CSTR (mean $\bar{T}_s$ only)	47.2	44.1	**90.6**	**48.2**	46.0	**92.0**	44.6	38.4	86.5
CSTR (mean $\bar{T}_s$ & Count)	46.5	42.8	89.8	47.9	44.6	91.4	41.6	37.0	84.1
Baseline (images only)	**47.3**	**44.8**	90.2	47.5	**46.6**	90.2	**47.9**	**40.3**	**91.1**

The best result per column is highlighted in **bold**.

The early-fusion results for the PKU-DDD17-CAR dataset, detailed in Table 6.9, present a different scenario. Despite the shorter event sampling duration (20 ms), providing less temporal context, the event-based signal is denser and spatially richer due to the camera being mounted on a moving vehicle. As a result, any ego-motion generates events around the edges of all objects in the scene, including the background, offering better texture information for object detection. Consequently, several models exhibit similar or superior generalization performance under certain metrics, as shown in Table 6.9. These include all spatio-temporal representations when evaluating combined test sequences and day-time sequences. However, surprisingly, the multi-modal approach underperforms compared to the frame-based baseline in low-light sequences. We attribute this to various factors, including data imbalance between both categories and the frame-based solution's more effective use of pre-trained weights for optimal convergence, unlike the multi-modal

approach.

Table 6.10 shows the results of applying the multi-modal augmentation framework using the early-fusion approach on the same dataset. Fewer multi-modal models outperform the frame-based baseline here. Notably, the CSTR, including only the mean timestamps, surpasses the baseline under the mAP_{50} metric overall, particularly in day-time sequences where both CSTR variants exceed the baseline under the mAP metric. This underscores the event data's potential in aiding object detection models to regress more accurate BBs with suitable event representations. Nonetheless, the multi-modal methods lag behind the baseline in low-light sequences, likely due to the limited number of low-light samples in the dataset, providing insufficient data for the models to learn effectively. The rich texture details in intensity images often suffice for localizing objects with low precision, potentially outweighing the benefits of event data, which might negatively influence a model under such conditions. However, the limitations of this dataset, as previously discussed, may constrain the full evaluation of this approach, emphasizing the need for diverse and large-scale multi-modal event-based datasets.

In summary, the early-fusion approach demonstrates the feasibility of leveraging the spatial similarities of frame-based and event-based modalities. A CNN employing 3D convolutional filters can extract more optimized low-level features from the channel-wise concatenation of these modalities. However, additional fine-tuning techniques are necessary to achieve better convergence when fusing both modalities. Lastly, the proposed augmentation framework has proven essential in enhancing the generalization capabilities of all object detection models utilizing the early fusion approach.

6.5.2.2 Late Fusion

The late fusion multi-modal object detection experiment results for MEVDT are presented in Tables 6.11 and 6.12. Despite the increased complexity of the overall model, our findings indicate that the late-fusion technique, as implemented in this work, generally underperforms compared to the early-fusion methods with this dataset. Further, these models consistently fall short of the

Table 6.11: Evaluation results of the late-fusion multi-modal method on MEVDT's test set. The results are demonstrated using different event representations across multiple batch-sampling durations.

Representation	**mAP**					**mAP$_{75}$ (%)**					**mAP$_{50}$ (%)**				
	43 ms	100 ms	200 ms	500 ms	**avg.**	43 ms	100 ms	200 ms	500 ms	**avg.**	43 ms	100 ms	200 ms	500 ms	**avg.**
Binary Event Frame	79.7	81.4	79.3	80.7	80.3	95.5	96.2	94.9	89.7	94.1	97.7	98.4	98.3	98.8	98.3
Polarized Event Frame	81.8	82.5	83.8	81.8	82.5	96.1	96.7	96.6	95.4	96.2	98.1	98.8	98.7	98.7	98.6
Binary Event Count	80.9	80.6	79.7	79.8	80.2	95.7	95.5	93.8	89.4	93.6	98.0	97.9	97.8	98.4	98.0
Polarized Event Count	81.2	82.0	83.2	80.4	81.7	95.8	96.3	94.8	91.7	94.7	98.1	98.3	98.4	98.5	98.3
Timestamp Image	81.6	84.2	84.7	84.6	83.8	96.2	96.2	97.4	96.9	96.7	98.0	98.2	98.5	**98.9**	98.4
Timestamp Image & Count	82.7	81.3	84.3	85.2	83.4	95.8	96.3	96.2	96.8	96.3	97.9	98.2	98.2	98.8	98.3
CSTR (mean $\bar{T}_s$ only)	83.9	83.1	81.9	85.1	83.5	96.2	95.9	96.6	96.8	96.4	98.3	97.9	98.6	98.9	98.4
CSTR (mean $\bar{T}_s$ & Count)	82.6	82.6	83.6	84.7	83.4	96.3	**97.2**	96.2	96.8	96.6	98.3	98.2	98.2	98.9	98.4
Baseline (images-only)	**88.2**	**88.5**	**86.6**	**88.4**	**87.9**	**97.9**	96.8	**97.9**	**97.8**	**97.6**	**98.9**	**98.9**	**99.0**	98.8	**98.9**

The best result per sampling duration and metric is in highlighted **bold**.

Table 6.12: Evaluation results of our late-fusion multi-modal approach in addition to the augmentation framework on MEVDT's test set. The results are demonstrated using different event representations across multiple batch-sampling durations.

Representation	**mAP**					**mAP$_{75}$ (%)**					**mAP$_{50}$ (%)**				
	43 ms	100 ms	200 ms	500 ms	**avg.**	43 ms	100 ms	200 ms	500 ms	**avg.**	43 ms	100 ms	200 ms	500 ms	**avg.**
Binary Event Frame	84.7	83.7	83.8	82.2	83.6	96.3	95.6	95.7	89.4	94.3	98.2	98.5	98.6	98.1	98.4
Polarized Event Frame	84.6	85.1	84.1	86.6	85.1	96.4	96.7	96.5	97.7	96.8	98.3	**98.7**	98.7	98.7	98.6
Binary Event Count	85.4	86.5	84.2	81.8	84.5	95.5	**97.0**	95.5	89.9	94.5	98.4	98.5	98.7	98.3	98.5
Polarized Event Count	85.6	86.3	84.0	82.8	84.7	96.5	95.5	96.1	88.4	94.1	**98.6**	98.3	98.5	98.4	98.5
Timestamp Image	84.8	85.9	85.2	87.5	85.8	96.3	96.6	**96.8**	97.6	96.8	98.2	98.5	98.7	98.6	98.5
Timestamp Image & Count	85.9	87.5	85.5	**88.2**	86.8	**97.2**	96.3	96.7	97.7	**97.0**	98.3	98.0	98.7	98.8	98.5
CSTR (mean $\bar{T}_s$ only)	84.4	85.4	85.2	86.2	85.3	96.3	95.8	96.7	96.4	96.3	98.2	**98.7**	98.7	98.8	98.6
CSTR (mean $\bar{T}_s$ & Count)	87.4	85.0	85.8	87.7	86.5	96.3	96.4	96.7	97.5	96.7	98.2	98.2	**98.8**	98.7	98.4
Baseline (images only)	**89.3**	**88.2**	**89.7**	86.4	**88.4**	96.4	96.7	96.7	**97.9**	96.9	98.4	**98.7**	98.7	**98.9**	**98.7**

The best result per sampling duration and metric is in highlighted **bold**.

performance achieved by the single-modal frame-based baseline models. The integration of our augmentation framework during training, as shown in Table 6.12, does enhance the performance of the late-fusion-based models. However, there remains a notable performance disparity with the frame-based baseline, particularly under the mAP metric.

The PKU-DDD17-CAR dataset yields similar trends. As evidenced in Table 6.13, the late fusion technique employed rarely surpasses the single-modal baseline. Additionally, the utilization of different event representations yields comparable performances. This outcome can be attributed to the dataset's short sampling duration and limited size. Incorporating our augmentation framework, as demonstrated in Table 6.14, yields intriguing results. Our straightforward late-fusion approach combining the Binary Event Frame and intensity images emerges as the top-performing model on this dataset across the mAP and mAP$_{75}$ metrics, even when compared to the early-fusion

Table 6.13: Evaluation results of the late-fusion multi-modal approach on PKU-DDD17-CAR's test set. The results are also demonstrated under daytime and low-light conditions.

Representation	**All Sequences**			**Day-Time Sequences**			**Low-Light Sequences**		
	mAP (%)	mAP_{75} (%)	mAP_{50} (%)	mAP (%)	mAP_{75} (%)	mAP_{50} (%)	mAP (%)	mAP_{75} (%)	mAP_{50} (%)
Binary Event Frame	43.2	40.3	84.0	43.6	42.1	84.0	41.7	32.8	84.4
Polarized Event Frame	43.2	38.9	84.7	44.0	40.7	84.9	40.5	32.8	85.5
Binary Event Count	43.3	41.0	84.2	43.8	42.7	84.1	42.0	35.0	85.8
Polarized Event Count	43.6	41.2	84.4	44.3	43.0	84.9	41.2	34.1	83.9
Timestamp Image	43.4	40.4	84.1	44.4	43.2	84.7	42.7	39.0	85.7
Timestamp Image & Count	43.6	41.3	84.3	44.2	43.0	84.4	41.6	33.9	85.1
CSTR (mean $\bar{T}_s$ only)	43.8	**41.8**	85.1	44.8	**44.3**	85.4	40.5	32.0	85.3
CSTR (mean $\bar{T}_s$ & Count)	43.6	41.0	84.3	44.8	43.9	85.2	39.9	31.1	82.2
Baseline (images only)	**44.5**	41.0	**87.1**	**44.9**	41.7	**87.5**	**43.5**	**39.1**	**86.1**

The best result per column is highlighted in **bold**.

Table 6.14: Evaluation results of the late-fusion multi-modal approach in addition to the proposed augmentation framework on PKU-DDD17-CAR's test set. The results are also demonstrated under daytime and low-light conditions.

Representation	**All Sequences**			**Day-Time Sequences**			**Low-Light Sequences**		
	mAP (%)	mAP_{75} (%)	mAP_{50} (%)	mAP (%)	mAP_{75} (%)	mAP_{50} (%)	mAP (%)	mAP_{75} (%)	mAP_{50} (%)
Binary Event Frame	**47.4**	**47.3**	87.2	**48.6**	**49.5**	87.6	45.0	**42.0**	87.3
Polarized Event Frame	45.8	44.3	86.4	46.9	47.4	86.5	43.7	37.1	87.2
Binary Event Count	45.1	43.2	86.2	46.3	46.2	86.5	42.7	35.7	87.4
Polarized Event Count	45.4	43.4	86.7	46.1	45.7	86.8	44.4	36.9	87.1
Timestamp Image	46.9	46.1	87.1	47.7	47.7	87.4	45.2	41.9	87.4
Timestamp Image & Count	45.2	42.6	85.8	45.6	44.0	85.9	44.6	39.7	86.4
CSTR (mean $\bar{T}_s$ only)	45.5	44.2	84.5	46.5	46.8	84.5	43.4	38.2	85.7
CSTR (mean $\bar{T}_s$ & Count)	45.2	41.1	86.3	45.7	43.3	86.0	44.7	35.6	88.6
Baseline (images only)	47.3	44.8	**90.2**	47.5	46.6	**90.2**	**47.9**	40.3	**91.1**

The best result per column is highlighted in **bold**.

approaches. While inherent limitations of this approach persist, the application of randomized augmentations has enabled each modality to achieve slightly improved convergence; leading the Binary Event Frame-based late fusion model to exhibit potential, despite the dataset's constraints. This underscores the necessity of large-scale datasets for a thorough evaluation and testing of event-based object detection solutions.

Overall, our experiments indicate that late fusion typically falls short of the performance achieved by single-modal frame-based and early-fusion multi-modal approaches. Although the models utilize the same optimizer and post-processing methods, their learning is indirect and implicit, lacking explicit inter-modality interactions. This can lead each modality's model to inadvertently converge to suboptimal solutions compared to when trained individually. A potentially more effective strategy might involve independently training each modality's model and subse-

quently integrating them through our late-fusion technique, followed by additional fine-tuning. While this approach adds complexity to the training process, it holds promise for yielding improved results. Moreover, the exploration of more intuitive post-processing techniques could further enhance model performance. In most scenarios, intensity images offer a more dependable signal. Therefore, dynamically adjusting the weighting between modalities, either on a local or global scale, could be pivotal for achieving superior multi-modal performance. This is particularly relevant under challenging lighting conditions, such as HDR or low-light environments, where the distinct advantages of each modality can be leveraged more effectively.

6.5.3 Augmentation Framework Ablation Study

In this section, we conduct an ablation study to examine the impact of various components of our augmentation framework. We focus on spatio-temporal representations for this study, considering their superior capabilities in encoding batches of event data compared to spatial representations. Additionally, we utilize the 500 ms sampling duration variant of the MEVDT dataset, along with the PKU-DDD17-CAR [95] dataset, to provide a comprehensive analysis. The results of this study are summarized in Table 6.15, showcasing the average performance per augmentation combination, dataset, and metric for both single-modal event-based and multi-modal fusion approaches. A detailed breakdown of the results for each representation across different modality configurations is available in Appendix D.

The augmentation framework proposed in this study significantly enhances performance compared to the baseline, as reflected in the test evaluation metrics. Spatial augmentations emerge as particularly crucial in object detection, outperforming any single augmentation method. This finding contrasts with our earlier results in Chapter 5 for the classification task, highlighting the importance of precise BB estimation in object detection. Spatial augmentations compel the model to learn translation-invariant features, which are essential for robust object detection. Other augmentation methods, like temporal augmentations, offer additional fine-tuning opportunities, reducing the model's reliance on irrelevant patterns. Temporal augmentations demonstrate superior perfor-

Table 6.15: Results of the augmentation ablation study using different modality configurations and augmentation-framework combinations. The average result of all four spatio-temporal representations is presented using each dataset's test set for each configuration and metric.

Fusion Method	**Augmentation Type**				**MEVDT** (ΔT = 500 ms)			**PKU-DDD17-CAR**		
	Spatial	Temporal	Polarity	Drop	mAP (%)	mAP_{75} (%)	mAP_{50} (%)	mAP (%)	mAP_{75} (%)	mAP_{50} (%)
Events Only	Baseline				80.8	94.2	97.4	21.7	18.4	45.2
				✓	81.8	94.1	97.6	22.2	18.6	46.0
			✓		81.4	93.1	97.1	21.6	18.6	45.2
		✓			82.9	94.8	98.0	22.0	18.9	45.5
		✓	✓		81.9	93.9	97.8	21.7	18.6	45.1
		✓	✓	✓	81.5	94.6	98.2	22.1	18.8	45.6
	✓				84.0	93.8	97.3	**24.0**	21.1	49.4
	✓	✓	✓		**84.8**	**94.9**	**98.7**	23.7	20.7	49.3
	✓	✓	✓	✓	84.2	94.1	98.4	23.9	**21.3**	**49.8**
Early Fusion	Baseline				86.9	96.9	**99.0**	44.7	40.3	87.5
				✓	86.5	96.9	98.9	44.2	39.4	87.6
			✓		85.4	96.9	98.9	44.1	39.5	87.4
		✓			86.2	96.9	98.9	44.7	40.4	87.6
		✓	✓		86.8	97.0	98.9	44.3	39.4	87.5
		✓	✓	✓	86.7	**97.2**	98.9	45.0	41.1	87.3
	✓				87.5	96.9	98.9	**47.6**	**44.4**	**91.2**
	✓	✓	✓		86.7	96.9	98.9	46.3	41.1	90.4
	✓	✓	✓	✓	**87.9**	**97.2**	98.9	46.2	41.9	89.8
Late Fusion	Baseline				84.9	96.8	**98.9**	43.6	41.1	84.4
				✓	84.6	97.1	98.7	43.9	41.5	83.6
			✓		84.4	96.4	98.7	43.7	40.8	84.7
		✓			83.9	97.2	98.8	43.8	41.4	83.6
		✓	✓		83.8	97.1	98.7	43.8	42.0	84.1
		✓	✓	✓	83.3	97.2	98.7	44.2	42.4	83.3
	✓				86.5	97.2	98.7	**46.1**	**45.6**	85.7
	✓	✓	✓		86.9	96.6	98.7	46.0	44.3	**86.3**
	✓	✓	✓	✓	**87.4**	**97.3**	98.7	45.7	43.5	85.9

The best results per metric and dataset are highlighted in **bold** for each fusion method.

mance compared to other event data-specific augmentations, such as polarity inversion and random event drop. Combining all four augmentation methods, as specified in Section 6.4.4, generally yields the best results using the MEVDT dataset for both fusion methods, and the PKU-DDD17-CAR dataset when employing a single-modal (event data only) approach.

Spatial augmentations are pivotal, providing essential variations that enhance the generalization capabilities of object detection models. While the other proposed augmentation methods are not indispensable, they contribute additional variations beneficial for improving the generalization of event-based models, particularly in single-modal scenarios. However, further experimentation with larger-scale multi-modal datasets and diverse augmentation settings is necessary to validate these findings and determine the optimal augmentation combination. Additionally, exploring larger ca-

pacity models, such as alternative one-stage models [102, 172], two-stage [152], and transformer-based architectures [27], will offer deeper insights into the interplay between model capacity and the efficacy of the proposed augmentation framework.

6.6 Conclusion

In this chapter, we present a comprehensive experimental study on the utilization of image-like event representations for object detection. Building upon the foundation set in Chapter 5, we evaluate the effectiveness of the Compact Spatio-Temporal Representation (CSTR) in comparison with other image-like representations for event-based and multi-modal object detection tasks. Additionally, we explore two multi-modal fusion techniques, early and late fusion, to examine the potential integration of frame-based and event-based vision, aiming to harness their combined strengths. Moreover, we propose a multi-modal augmentation framework specifically tailored for event-based and frame-based modalities. The methods proposed in this chapter are rigorously tested on two distinct event-based multi-modal datasets, offering a comprehensive analysis of their performance.

Our investigation underscores the importance of selecting optimal spatio-temporal representations, such as the CSTR, for enhancing event-based vision tasks. These representations are crucial for enabling direct compatibility with advanced computer vision architectures and effectively leveraging their pre-trained weights. However, it is noteworthy that in our experimental setup, traditional frame-based methods generally outperform event-based and multi-modal solutions. This limitation is largely attributed to the datasets used, which do not present sufficiently challenging scenarios to fully harness the unique capabilities of event-based sensors.

Looking forward, we identify several avenues for future research on event-based object detection:

- Investigating the application of the presented methodologies on more sophisticated and diverse object detection networks, such as RetinaNet [102].

- Developing and leveraging specialized datasets that encompass a wide range of challenging visual environments with long sequence durations, thereby enabling the effective use of temporally-aware and memory-based solutions.
- Examining feature-level fusion techniques for potentially more optimal integration of frame-based and event-based modalities.
- Exploring varied fine-tuning strategies for object detection models, especially for architectures like SSD, to optimize their adaptation to event-based data.
- Extending the testing to larger multi-modal datasets as they become available.
- Implementing advanced methods for dynamically weighting modalities based on specific scene conditions, both globally and locally, within the scene.

In conclusion, this chapter not only presents a detailed study in the field of event-based object detection but also sets the stage for further exploration. The insights gained here set the path for future research efforts aimed at fully capitalizing on the unique advantages of event-based and multi-modal vision systems in varied and complex real-world contexts.

CHAPTER 7

Conclusions

Event-based vision, a novel and emerging field in visual sensing technology, represents a significant paradigm shift from traditional frame-based imaging techniques. Characterized by asynchronous data capturing, high temporal resolution, and exceptional HDR capabilities, event-based sensors offer a promising alternative to conventional systems. These sensors, inspired by the human eye's retina, operate asynchronously at the pixel level, detecting and recording changes in intensity independently. This unique approach allows for the continuous monitoring of a scene, capturing information only when and where it is needed.

The inherent advantages of event-based vision, such as its high temporal resolution and HDR, position it as a robust solution for dynamic and challenging environments. These sensors excel in scenarios where speed and responsiveness are crucial, effectively eliminating motion blur issues common in traditional cameras and providing detailed imagery in scenes with both bright and dark areas. Moreover, their low latency and power consumption make them highly suitable for time-sensitive applications like AVs and ADAS.

Despite these strengths, integrating event-based sensors into established CV architectures presents significant challenges. The asynchronous and sparse nature of their output differs substantially from the synchronous, dense inputs of traditional systems. This dissertation has aimed to address these challenges, developing methodologies that leverage the unique strengths of event-based vision while overcoming its inherent limitations. The research presented here not only addresses various technical challenges but also demonstrates the practical applicability of event-

based vision in robotic perception tasks, contributing valuable insights and laying a foundation for ongoing research in the field.

7.1 Dissertation Summary

This dissertation presents a journey through the evolving field of event-based vision, focusing on leveraging this novel modality in various robotic perception tasks. Each chapter contributes uniquely to this domain, showcasing diverse methodologies and significant advancements. The contributions of this work are as follows:

1. Multi-Modal Event-Based Vehicle Detection and Tracking Dataset (Chapter 2): Introduction of the MEVDT dataset, a pivotal resource supporting research in event-based object detection and tracking. This dataset, with its synchronized streams of event data and grayscale images, provides a unique platform for developing and benchmarking novel algorithms in automotive environments.

2. Hybrid Approach for High-Temporal-Resolution Object Detection and Tracking (Chapter 3): Development of a hybrid methodology combining frame-based object detectors with novel event-based methods for improved detection and tracking. This approach showcases the potential of integrating asynchronous event data with conventional frame-based methods for enhanced temporal resolution and accuracy.

3. Advanced Techniques in Event-Based Vehicle Detection and Tracking (Chapter 4): Introduction of advanced event-based techniques to refine detection accuracy and tracking robustness. These methods demonstrate significant improvements in tracking performance, particularly when utilizing small-scale real-time object detectors.

4. Development of the Compact Spatio-Temporal Representation (CSTR) (Chapter 5): Creation of the CSTR, a novel representation for encoding event data compatible with modern

CV architectures. The CSTR's ability to efficiently encapsulate spatial, temporal, and polarity information enhances its efficacy in various recognition tasks, bridging the gap between event-based data and traditional CV methods.

5. Evaluation of Image-Like Event Representations for Object Detection (Chapter 6): Comprehensive examination of the application of the CSTR and other image-like event representations in object detection. This chapter not only assesses the comparative performance of these representations but also explores their integration in multi-modal object detection, providing insights into the synergies between different sensing modalities.

7.2 Future Work

The collective contributions of this dissertation have led to the advancement of the field of event-based vision and its application in CV, particularly in robotic perception tasks. The methodologies developed and findings obtained have not only addressed some of the current challenges but also opened new avenues for exploration and application.

Future research should focus on bridging the gaps between these methodologies and exploring new applications of event-based vision. There is a promising avenue in applying these findings in real-world scenarios, where the unique advantages of event-based sensors—such as their high dynamic range and immunity to motion blur—can be fully leveraged. Potential applications could include autonomous navigation in diverse and challenging environments, advanced surveillance systems, and enhanced human-robot interaction. The integration of event-based vision into the robotic perception stack offers the possibility of more robust, efficient, and adaptive systems.

Building upon these broader perspectives, specific future research directions emerging from each chapter of this dissertation include:

- Expansion and Diversification of Event-Based Datasets (Chapter 2): Enhancing the MEVDT dataset to include dynamic scenes and more pedestrian data. This expansion will enable the

dataset to better represent real-world scenarios, thus broadening its applicability and utility in event-based vision research.

- Advanced Association Metrics and Fully Event-Based Approaches (Chapter 3): Exploring sophisticated metrics and fully event-based methodologies for object detection and tracking. This direction could offer more dynamic and robust solutions, especially for objects with rapidly changing shapes.

- Integration of Learned Event-Based Methods (Chapter 4): Replacing some classical and hand-crafted components of the presented framework with learned event-based methods to achieve more dynamic performance under challenging scenarios. This approach requires larger and more sophisticated labeled event-based datasets.

- Robustness Enhancement and Noise Mitigation for CSTR (Chapter 5): Optimizing the CSTR for real-time applications and enhancing its robustness against sensor noise. Evaluating the CSTR's suitability for real-time applications where latency is a concern would also be a valuable direction.

- Sophisticated Object Detection Networks and Specialized Datasets (Chapter 6): Applying methodologies on advanced object detection networks and developing specialized datasets for temporally-aware solutions. This direction involves creating datasets that cover a wide range of challenging visual environments and long sequence durations.

- Dynamic Weighting of Modalities and Larger Multi-Modal Datasets: Implementing dynamic weighting strategies for modalities and extending testing to larger datasets. This approach aims to enhance the performance and applicability of event-based vision systems in complex real-world contexts.

These specific directions aim to build upon the solid foundation established in this dissertation, exploring new frontiers in event-based vision. They promise to address current limitations and

open up exciting possibilities for the operationalization of event-based vision in a wide range of applications.

APPENDIX A

Supplementary Tables for Chapter 2

This appendix provides detailed supplementary tables that expand upon the dataset structure and statistics discussed in Chapter 2. These tables offer an in-depth view of the dataset's composition, with a focus on the individual sequences within Scenes A and B.

Table A.1 presents a comprehensive breakdown for each sequence in Scene A. This includes detailed information on sequence duration, the number of images, events, objects, and the average area of bounding boxes. It also specifies the allocation of each sequence to either the training or testing splits. This level of detail is crucial for understanding the structure and distribution of the dataset, particularly how it is divided for machine learning purposes.

Similarly, Table A.2 provides an analogous breakdown for Scene B. It gives an exhaustive overview of each sequence, covering all relevant parameters and classifications. This table is integral for a complete understanding of Scene B's dataset structure, ensuring transparency and clarity in how the data is processed and utilized in the research.

These supplementary tables are essential for readers seeking a granular understanding of the dataset and its composition. They also serve as a valuable resource for replicating the research or for further exploration into event-based vision systems.

Table A.1: Detailed sequence statistics for Scene A. The table includes information on each sequence's duration, number of images, events, objects, average bounding box area, and its allocation to training or testing splits, offering an in-depth view of the dataset's composition in Scene A.

Seq. #	**Seq. Name**	**Duration** (s)	**# of Images**	**# of Events**	**# of Objects**	**Avg. Bounding Box Area** ($pixel^2$)	**Train | Test**
1	1581956305832790936	9.5	222	76924	240	1852.4	Train
2	1581956366514475936	21.2	494	48556	122	1562.3	Train
3	1581956422501835936	10.4	243	70138	241	1490.3	Test
4	1581956475991297936	23.3	542	61664	135	1754.2	Test
5	1581956525690846936	17.4	404	79738	382	1476.3	Train
6	1581956568112038936	5.3	124	34207	117	1394.1	Test
7	1581956586329463936	12.1	283	114163	186	2997.0	Train
8	1581956636804222936	4.9	115	29212	102	1522.7	Train
9	1581956672808401936	5.4	127	60633	118	1665.5	Train
10	1581957068983574936	21.0	488	154064	420	3373.5	Train
11	1581957114204134936	7.0	163	34310	160	1609.8	Train
12	1581957156969863936	8.3	195	62531	192	4538.2	Test
13	1581957173378467936	2.5	59	20295	58	1315.7	Train
14	1581957190648414936	45.6	1061	107768	224	1796.1	Train
15	1581957249133671936	6.7	158	51306	150	1730.8	Train
16	1581957506675527936	18.1	421	77074	502	1373.6	Train
17	1581957567314145936	4.1	96	34093	81	1710.7	Train
18	1581957616841425936	10.3	241	41452	208	1386.9	Train
19	1581957903798179936	6.2	145	31237	130	1658.5	Train
20	1581957963058646936	5.2	124	48102	122	2071.8	Train
21	1581958023266591936	4.6	109	20877	97	1476.4	Train
22	1581958094284404936	5.1	119	14818	113	1587.9	Train
23	1581958106816959936	14.9	348	140138	399	1396.0	Train
24	1581958201263329936	25.4	592	91349	239	1737.3	Train
25	1581958201392531936	2.8	65	23145	64	1608.6	Train
26	1581958289206551936	14.9	346	71577	282	1816.2	Train
27	1581958320817876936	5.4	126	76807	113	5773.9	Train
28	1581958380465948936	29.7	694	122720	578	1487.0	Test
29	1581958511820908936	9.7	226	78420	198	1541.7	Train
30	1581958540632865936	3.9	91	33334	84	1822.8	Test
31	1581958551959539936	29.0	676	330350	605	2782.1	Train
32	1581958587877583936	7.6	177	28911	166	1426.3	Train

Table A.2: Detailed sequence statistics for Scene B. This table presents comprehensive data for each sequence in Scene B, including duration, image count, event count, object count, average bounding box area, and their distribution in training and testing, providing a thorough insight into Scene B's dataset structure.

Seq. #	**Seq. Name**	**Duration** (s)	**# of Images**	**# of Events**	**# of Objects**	**Avg. Bounding Box Area** ($pixel^2$)	**Train | Test**
1	1603470885671858364	14.1	329	130227	321	5706.2	Test
2	1603470907722265364	4.9	116	109775	107	4216.0	Train
3	1603470947042618364	8.3	195	106234	188	2468.5	Train
4	1603471304371177364	5.8	137	123331	90	3988.2	Train
5	1603471325344903364	2.1	49	87050	44	3017.5	Train
6	1603471347223041364	2.7	65	106671	55	3995.4	Train
7	1603471362511897364	2.4	58	91918	47	3043.1	Train
8	1603471387318604364	2.5	61	108368	52	3971.9	Train
9	1603471400411033364	2.1	51	64688	34	2924.9	Test
10	1603471419705138364	6.0	142	91522	127	5007.8	Train
11	1603471437405757364	1.7	41	55782	27	3028.5	Train
12	1603471457905745364	6.8	159	92745	142	4610.4	Train
13	1603471475606364364	2.4	56	67010	40	3087.3	Train
14	1603471489904674364	2.9	68	151037	52	4502.9	Train
15	1603471504116850364	6.9	163	94593	152	5330.4	Train
16	1603471523712426364	2.3	56	67044	45	2963.8	Test
17	1603471544513884364	17.6	410	104188	400	5260.7	Train
18	1603471574445588364	2.3	56	64461	43	3054.5	Train
19	1603471594816373364	6.8	159	105311	150	5210.7	Train
20	1603471817884627727	2.8	67	83430	53	4643.3	Train
21	1603471844801627727	3.8	91	116064	76	4082.0	Test
22	1603471863492792727	4.1	96	216927	76	7410.9	Train
23	1603471880891941727	3.6	86	103624	64	3000.1	Train
24	1603471908885621727	6.4	151	112285	142	5159.1	Train
25	1603471928136659727	2.5	61	72175	49	3057.4	Train
26	1603471965045249727	6.8	161	119105	151	5128.6	Train
27	1603471985975909727	3.1	73	87510	47	3042.6	Train
28	1603472011687027727	2.7	65	113585	51	3972.8	Test
29	1603472029387646727	2.8	68	74378	57	3059.0	Train
30	1603472052643934727	6.2	146	126208	136	5031.5	Test
31	1603472118493683727	2.0	49	148406	45	3913.5	Train

APPENDIX B

Supplementary Tables for Chapter 3

In this appendix, we provide a supplementary table that complements the results and discussions presented in Chapter 3. This table offers a detailed view of the data and findings related to our high-temporal-resolution object detection and tracking research.

Table B.1 contains a comprehensive breakdown of the total number of object IDs and detections for both the ground truth data and the prediction results from various tracking configurations used in our evaluation. This level of granularity is vital for a thorough understanding of how the tracking rate and mode impact the number of detections and the behavior of the system under different temporal resolutions.

The table includes information for both the frame-based object detectors, SSD [108] and YOLOv3 [149], across various tracking rates and modes. It details the total number of unique object ID trajectories (85 in our dataset) and shows how the number of ground truth detections scales with the temporal resolution. In both cases, the number of IDs and detections resulting from each tracking configuration must be as close as possible to the ground truth statistics at each tracking rate. This data is crucial for evaluating the performance of our proposed methods, especially in terms of object tracking consistency and detection accuracy at different temporal resolutions.

These supplementary details are needed for readers interested in the technical nuances of our tracking approaches and for those seeking to replicate or build upon our research. They provide additional details to conclusions drawn in Chapter 3, highlighting the effectiveness of different tracking modes and their impact on object detection and tracking performance.

Table B.1: The total number of IDs and detections for both the ground truth data and the predicted results for the different tracking configurations used in our evaluation. Our dataset has a total number of unique object ID trajectories of 85. The number of ground truth detections increases as the temporal resolution of the data increases.

Tracking Rate	**Tracking Mode**	**Ground Truth**		**SSD**		**YOLOv3**	
		IDs	**Dets**	**IDs**	**Dets**	**IDs**	**Dets**
24 Hz	*	85	9891	110	7723	105	6462
48 Hz	1	85	19,777	125	14,908	117	12,480
	2			125	15,405	117	12,899
	3			125	15,155	117	12,710
96 Hz	1	85	39,549	147	28,427	139	23,748
	2			147	30,773	139	25,773
	3			147	30,022	139	25,207
192 Hz	1	85	79,093	149	51,226	142	42,332
	2			147	61,498	139	51,521
	3			147	59,720	139	50,179
384 Hz	1	85	158,181	171	82,708	172	66,862
	2			165	122,972	165	103,023
	3			165	119,131	165	100,131

* Image-only tracking (excludes event data).

APPENDIX C

Supplementary Tables for Chapter 5

This appendix complements the findings presented in Chapter 5, particularly the results of the two key experiments conducted as part of our research. These supplementary tables provide a thorough breakdown of the performance metrics and insights that could not be fully captured within the main body of the chapter due to space constraints.

Table C.1 offers an extensive analysis of the baseline evaluation results, as referenced in Table 5.2 in Chapter 5. It details the performance of each of the six pre-trained classification networks (specified in Section 5.4.1.2) across different event-based recognition datasets. This table allows for an in-depth comparison of various network configurations and their effectiveness with different types of image-like event representations. The granularity of this data is instrumental in understanding the nuanced performance differences among these configurations and how they relate to the foundational principles of our research.

Similarly, Table C.2 presents an expanded view of the results from our augmentation-framework evaluation shown in Table 5.3. This table breaks down the performance of the top three classification networks across different spatio-temporal representations, augmentation types, and datasets. The detailed results enable a deeper understanding of how various augmentations influence classification accuracy and the robustness of different spatio-temporal representations. This is particularly relevant for assessing the effectiveness of our proposed event-based augmentation framework and its impact on the Compact Spatio-Temporal Representation (CSTR) and other representations.

These tables are vital for readers seeking a comprehensive understanding of the experimental

Table C.1: A full breakdown of the test classification accuracy results that are presented in Table 5.2 for the event-based recognition datasets. The results of each evaluated representation configuration are demonstrated for 6 different pre-trained classification networks that are fine-tuned on each dataset.

Representation	**Pre-trained Classifier**	**Dataset** (# of channels)																	
		N-MNIST			N-Cars			N-Caltech101			CIFAR10-DVS			ASL-DVS			DVS-Gesture		
		(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
Binary Event Frame	ResNet18	95.5%	95.0%	95.0%	92.6%	91.7%	93.0%	71.9%	64.8%	74.8%	61.4%	60.2%	61.3%	99.7%	99.3%	99.7%	87.3%	84.1%	85.4%
	ResNet50	95.1%	95.0%	95.3%	90.3%	90.5%	90.0%	69.1%	67.9%	74.7%	53.4%	49.6%	52.7%	99.5%	99.7%	**99.9%**	81.6%	82.9%	81.5%
	MobileNetV2	95.0%	94.6%	95.6%	92.1%	**94.3%**	94.4%	67.3%	68.3%	72.6%	48.2%	29.5%	48.6%	99.4%	99.6%	99.5%	82.5%	86.3%	84.3%
	MobileNetV3-L	95.0%	95.2%	95.3%	93.1%	88.5%	90.4%	69.7%	66.7%	71.1%	43.6%	51.1%	49.2%	99.6%	98.6%	99.7%	86.0%	84.7%	84.6%
	MobileNetV3-S	94.9%	95.2%	95.1%	90.2%	90.6%	93.1%	64.5%	64.0%	71.2%	46.7%	28.9%	45.9%	99.5%	95.6%	99.7%	78.7%	81.9%	83.5%
	InceptionV3	-	-	94.7%	-	-	94.5%	-	-	76.8%	-	-	59.1%	-	-	99.8%	-	-	85.8%
Polarized Event Frame	ResNet18	98.8%	96.7%	99.2%	93.2%	89.6%	94.0%	**80.8%**	72.6%	78.4%	67.8%	72.4%	68.3%	99.7%	**99.9%**	**99.9%**	90.4%	91.6%	91.8%
	ResNet50	98.8%	96.3%	99.0%	88.8%	90.9%	89.2%	77.1%	72.6%	84.5%	60.4%	64.4%	58.8%	**99.8%**	**99.9%**	**99.9%**	90.0%	90.9%	90.1%
	MobileNetV2	**99.1%**	95.9%	99.0%	92.6%	93.0%	94.0%	76.0%	69.0%	80.0%	45.7%	57.0%	55.4%	99.7%	99.8%	99.5%	88.5%	91.0%	88.5%
	MobileNetV3-L	98.6%	96.3%	98.2%	**95.0%**	76.1%	92.7%	74.2%	67.6%	80.7%	58.7%	59.1%	57.3%	99.2%	99.6%	99.6%	89.3%	88.7%	91.4%
	MobileNetV3-S	98.8%	95.2%	99.0%	90.3%	92.6%	94.5%	71.8%	67.4%	79.8%	53.2%	57.0%	57.8%	99.7%	99.4%	99.4%	88.3%	90.8%	90.5%
	InceptionV3	-	-	99.1%	-	-	95.1%	-	-	85.4%	-	-	67.0%	-	-	99.8%	-	-	91.3%
Binary Event Count	ResNet18	98.4%	98.5%	98.5%	89.3%	93.0%	91.3%	79.4%	76.9%	81.1%	**79.4%**	**78.3%**	78.7%	90.4%	53.0%	59.2%	84.7%	89.0%	86.3%
	ResNet50	98.8%	98.8%	98.7%	92.6%	91.3%	92.2%	78.0%	72.6%	80.8%	72.8%	70.0%	75.1%	85.3%	52.6%	61.8%	88.8%	85.0%	91.2%
	MobileNetV2	98.8%	98.7%	98.5%	93.0%	93.0%	93.3%	74.2%	74.8%	81.4%	65.4%	66.2%	67.1%	5.5%	35.4%	90.8%	**91.1%**	89.7%	90.2%
	MobileNetV3-L	98.3%	98.5%	98.3%	90.9%	90.9%	90.4%	76.4%	69.7%	80.1%	64.0%	70.6%	70.3%	39.5%	9.5%	76.3%	88.6%	88.3%	90.9%
	MobileNetV3-S	98.6%	98.7%	98.1%	92.0%	93.8%	87.8%	67.9%	71.9%	78.4%	64.6%	67.2%	71.6%	8.3%	9.0%	86.2%	85.0%	83.7%	84.1%
	InceptionV3	-	-	99.1%	-	-	91.6%	-	-	84.9%	-	-	**79.5%**	-	-	95.7%	-	-	91.9%
Polarized Event Count	ResNet18	-	99.1%	98.9%	-	92.5%	92.9%	-	75.7%	82.0%	-	77.5%	79.3%	-	17.1%	8.6%	-	92.1%	91.4%
	ResNet50	-	98.9%	97.6%	-	91.2%	92.8%	-	73.0%	82.2%	-	73.4%	72.7%	-	28.9%	55.4%	-	92.8%	90.5%
	MobileNetV2	-	98.9%	98.9%	-	92.4%	94.1%	-	73.8%	80.7%	-	64.1%	66.6%	-	41.8%	72.5%	-	90.6%	92.2%
	MobileNetV3-L	-	98.9%	98.7%	-	90.4%	92.6%	-	72.4%	79.5%	-	66.4%	65.2%	-	67.7%	54.8%	-	92.1%	93.0%
	MobileNetV3-S	-	98.7%	98.0%	-	92.4%	92.2%	-	70.4%	79.6%	-	68.2%	73.0%	-	49.7%	63.6%	-	91.2%	91.6%
	InceptionV3	-	-	99.1%	-	-	92.4%	-	-	85.9%	-	-	74.0%	-	-	56.9%	-	-	86.9%
Timestamp Image	ResNet18	-	99.0%	99.0%	-	86.9%	91.8%	-	73.1%	82.3%	-	74.7%	76.4%	-	99.8%	99.5%	-	91.6%	92.9%
	ResNet50	-	99.1%	99.1%	-	92.1%	94.8%	-	76.8%	80.4%	-	67.8%	67.6%	-	**99.9%**	**99.9%**	-	89.9%	93.5%
	MobileNetV2	-	98.9%	99.1%	-	83.2%	95.3%	-	75.1%	82.0%	-	63.5%	65.1%	-	99.7%	**99.9%**	-	92.3%	92.5%
	MobileNetV3-L	-	99.0%	98.6%	-	90.9%	85.5%	-	73.0%	80.5%	-	67.4%	62.3%	-	99.4%	99.8%	-	92.8%	92.8%
	MobileNetV3-S	-	98.9%	98.7%	-	74.5%	92.5%	-	72.7%	79.7%	-	65.4%	69.1%	-	98.5%	99.3%	-	90.4%	92.9%
	InceptionV3	-	-	**99.3%**	-	-	93.6%	-	-	83.2%	-	-	72.3%	-	-	99.8%	-	-	**94.7%**
Timestamp Image & Count	ResNet18	-	-	99.0%	-	-	92.2%	-	-	84.4%	-	-	78.7%	-	-	99.8%	-	-	93.6%
	ResNet50	-	-	99.1%	-	-	92.6%	-	-	82.7%	-	-	71.6%	-	-	**99.9%**	-	-	91.2%
	MobileNetV2	-	-	98.7%	-	-	93.1%	-	-	82.0%	-	-	66.8%	-	-	99.6%	-	-	91.7%
	MobileNetV3-L	-	-	98.7%	-	-	89.8%	-	-	82.1%	-	-	70.1%	-	-	99.7%	-	-	92.4%
	MobileNetV3-S	-	-	98.7%	-	-	90.3%	-	-	77.4%	-	-	72.3%	-	-	99.7%	-	-	94.0%
	InceptionV3	-	-	99.1%	-	-	95.1%	-	-	86.5%	-	-	76.2%	-	-	99.4%	-	-	94.3%
CSTR (mean $\bar{T}_s$ only)	ResNet18	-	**99.3%**	99.2%	-	93.1%	92.7%	-	**80.8%**	84.9%	-	74.2%	74.2%	-	**99.9%**	99.8%	-	93.9%	92.6%
	ResNet50	-	99.2%	99.1%	-	92.4%	93.7%	-	79.6%	84.0%	-	63.8%	67.3%	-	99.8%	99.8%	-	92.5%	93.4%
	MobileNetV2	-	99.1%	99.3%	-	93.6%	92.5%	-	77.9%	85.0%	-	59.5%	61.7%	-	99.8%	**99.9%**	-	**94.2%**	94.5%
	MobileNetV3-L	-	99.0%	98.5%	-	90.8%	92.9%	-	73.8%	83.0%	-	63.6%	63.2%	-	99.0%	98.8%	-	91.8%	93.0%
	MobileNetV3-S	-	98.0%	98.9%	-	92.0%	90.7%	-	72.5%	81.6%	-	56.7%	64.6%	-	99.6%	99.4%	-	91.8%	93.5%
	InceptionV3	-	-	99.2%	-	-	91.7%	-	-	85.0%	-	-	73.1%	-	-	99.8%	-	-	94.5%
CSTR (mean $\bar{T}_s$ & Count)	ResNet18	-	-	99.1%	-	-	93.0%	-	-	81.6%	-	-	77.8%	-	-	**99.9%**	-	-	92.8%
	ResNet50	-	-	99.2%	-	-	92.5%	-	-	85.4%	-	-	70.6%	-	-	**99.9%**	-	-	94.2%
	MobileNetV2	-	-	99.2%	-	-	**95.6%**	-	-	83.0%	-	-	65.2%	-	-	99.8%	-	-	94.6%
	MobileNetV3-L	-	-	98.9%	-	-	93.6%	-	-	82.2%	-	-	65.9%	-	-	99.1%	-	-	93.5%
	MobileNetV3-S	-	-	98.8%	-	-	93.6%	-	-	77.9%	-	-	71.0%	-	-	99.7%	-	-	93.1%
	InceptionV3	-	-	99.2%	-	-	93.5%	-	-	**87.7%**	-	-	79.0%	-	-	**99.9%**	-	-	93.3%

The best values per dataset and number of input channels are highlighted in **bold**.

outcomes and for those interested in replicating or extending this research. They offer a complete picture of our findings, highlighting the strengths and limitations of different event representations explored in our work.

Table C.2: A full breakdown of the test classification accuracy results that are presented in Table 5.3 for the event-based recognition datasets. The results of each evaluated spatio-temporal representation, in combination with the augmentation framework, are demonstrated for the top-3 pre-trained classification networks fine-tuned on each dataset.

Representation	**Augmentation Type**			**Classifier**	**Dataset**				
	Spatial	Temporal	Polarity		N-MNIST	N-Cars	N-Caltech101	CIFAR10-DVS	DVS-Gesture
Timestamp Image		Baseline		ResNet18	99.0%	91.8%	82.3%	76.4%	92.9%
				ResNet50	99.1%	94.8%	80.4%	67.6%	93.5%
				InceptionV3	99.3%	93.6%	83.2%	72.3%	94.7%
	✓			ResNet18	99.4%	94.6%	81.6%	76.4%	94.2%
				ResNet50	99.3%	93.9%	85.8%	76.6%	94.2%
				InceptionV3	99.3%	95.1%	85.8%	79.5%	93.9%
		✓		ResNet18	99.1%	96.1%	85.3%	74.9%	93.3%
				ResNet50	99.3%	93.7%	88.2%	73.0%	95.3%
				InceptionV3	99.2%	97.2%	87.7%	80.4%	94.4%
			✓	ResNet18	99.2%	96.0%	83.7%	71.9%	92.9%
				ResNet50	99.0%	95.5%	85.9%	68.5%	94.5%
				InceptionV3	99.2%	95.1%	89.0%	75.2%	94.1%
		✓	✓	ResNet18	99.2%	93.9%	83.6%	75.3%	93.0%
				ResNet50	99.2%	96.2%	88.9%	74.8%	94.0%
				InceptionV3	99.1%	**97.4%**	88.1%	78.8%	94.6%
	✓	✓	✓	ResNet18	99.3%	95.9%	82.6%	78.0%	94.5%
				ResNet50	99.1%	96.3%	86.5%	76.5%	95.0%
				InceptionV3	99.2%	96.8%	86.5%	79.7%	95.2%
Timestamp Image & Count		Baseline		ResNet18	99.0%	92.2%	84.4%	78.7%	93.6%
				ResNet50	99.1%	92.6%	82.7%	71.6%	91.2%
				InceptionV3	99.1%	95.1%	86.5%	76.2%	94.3%
	✓			ResNet18	99.4%	96.3%	82.2%	80.0%	94.4%
				ResNet50	99.4%	94.7%	84.9%	78.5%	94.3%
				InceptionV3	99.4%	96.2%	86.1%	82.6%	95.1%
		✓		ResNet18	99.0%	95.5%	84.8%	76.4%	94.6%
				ResNet50	99.2%	95.5%	87.4%	74.6%	94.6%
				InceptionV3	99.4%	95.1%	89.3%	80.7%	94.6%
			✓	ResNet18	99.4%	96.1%	84.2%	74.0%	91.4%
				ResNet50	99.0%	95.6%	86.4%	69.7%	93.6%
				InceptionV3	99.3%	97.1%	88.8%	76.9%	94.9%
		✓	✓	ResNet18	99.2%	95.9%	86.2%	76.9%	94.2%
				ResNet50	99.2%	94.2%	86.3%	76.6%	94.0%
				InceptionV3	99.4%	96.9%	89.2%	81.0%	95.0%
	✓	✓	✓	ResNet18	99.3%	96.9%	84.3%	79.4%	94.0%
				ResNet50	99.2%	95.1%	87.5%	78.6%	95.2%
				InceptionV3	99.4%	97.1%	87.7%	**82.8%**	94.1%

The best values per dataset are highlighted in **bold**.

Table C.2: A full breakdown of the test classification accuracy results that are presented in Table 5.3 for the event-based recognition datasets. The results of each evaluated spatio-temporal representation, in combination with the augmentation framework, are demonstrated for the top-3 pre-trained classification networks fine-tuned on each dataset. (***Continued***)

Representation	**Augmentation Type**			**Classifier**	**Dataset**				
	Spatial	Temporal	Polarity		N-MNIST	N-Cars	N-Caltech101	CIFAR10-DVS	DVS-Gesture
CSTR (mean $\bar{T}_s$ only)		Baseline		ResNet18	99.2%	92.7%	84.9%	74.2%	92.6%
				ResNet50	99.1%	93.7%	84.0%	67.3%	93.4%
				InceptionV3	99.2%	91.7%	85.0%	73.1%	94.5%
	✓			ResNet18	99.4%	95.7%	83.8%	75.2%	94.5%
				ResNet50	99.3%	95.5%	85.3%	73.6%	96.1%
				InceptionV3	**99.5%**	97.1%	88.0%	78.2%	95.5%
		✓		ResNet18	99.3%	95.1%	85.8%	75.9%	92.3%
				ResNet50	99.4%	96.1%	88.8%	73.8%	94.0%
				InceptionV3	99.3%	88.6%	88.1%	76.7%	93.9%
			✓	ResNet18	99.4%	96.3%	86.5%	70.4%	94.5%
				ResNet50	**99.5%**	96.3%	87.9%	66.7%	95.8%
				InceptionV3	99.2%	96.1%	89.0%	75.5%	94.1%
		✓	✓	ResNet18	99.3%	97.3%	85.8%	75.4%	93.8%
				ResNet50	99.2%	96.2%	89.0%	73.5%	92.6%
				InceptionV3	99.1%	97.1%	**90.1%**	75.5%	94.8%
	✓	✓	✓	ResNet18	99.3%	96.4%	84.9%	77.4%	93.5%
				ResNet50	99.4%	96.4%	86.3%	77.5%	96.0%
				InceptionV3	99.2%	97.0%	88.1%	79.9%	95.6%
CSTR (mean $\bar{T}_s$ & Count)		Baseline		ResNet18	99.1%	93.0%	81.6%	77.8%	92.8%
				ResNet50	99.2%	92.5%	85.4%	70.6%	94.2%
				InceptionV3	99.2%	93.5%	87.7%	79.0%	93.3%
	✓			ResNet18	99.4%	95.9%	81.5%	78.6%	94.5%
				ResNet50	99.4%	96.0%	86.6%	78.0%	96.1%
				InceptionV3	**99.5%**	96.9%	86.8%	81.3%	**96.5%**
		✓		ResNet18	99.3%	94.4%	85.6%	78.2%	94.5%
				ResNet50	99.4%	96.2%	88.4%	76.6%	94.8%
				InceptionV3	99.4%	95.5%	89.8%	80.4%	95.3%
			✓	ResNet18	99.1%	96.4%	86.1%	73.1%	95.3%
				ResNet50	99.4%	94.8%	86.7%	66.8%	95.5%
				InceptionV3	99.3%	97.1%	88.4%	76.9%	94.4%
		✓	✓	ResNet18	99.3%	96.6%	86.7%	77.9%	93.4%
				ResNet50	99.4%	96.2%	88.6%	75.4%	95.4%
				InceptionV3	99.4%	96.9%	89.8%	80.4%	94.5%
	✓	✓	✓	ResNet18	99.3%	96.9%	84.0%	78.8%	95.1%
				ResNet50	**99.5%**	96.9%	86.2%	78.7%	95.7%
				InceptionV3	99.3%	97.2%	88.2%	81.8%	96.3%

The best values per dataset are highlighted in **bold**.

APPENDIX D

Supplementary Tables for Chapter 6

This appendix provides a detailed extension of the ablation study presented in Chapter 6, specifically elaborating on the results in Table 6.15. The main aim of this study is to investigate the effects of different augmentation components on spatio-temporal representations, particularly for single-modal event-based and multimodal fusion approaches. The study primarily utilizes the 500 ms sampling duration variant of the MEVDT dataset and the PKU-DDD17-CAR dataset as described in Chapter 6, offering a varying basis for assessing the performance of single-modal event-based and multimodal fusion methods under varied randomized augmentation conditions.

The tables included here offer an in-depth view of the results, dissecting the impact of each augmentation configuration across various spatio-temporal representations. This level of detail is crucial for thoroughly understanding how different augmentations, both individually and in combination, influence the performance metrics of the object detection models.

- Table D.1 demonstrates the effects of various augmentation configurations using the event-based vision modality. It presents the results for each spatio-temporal representation, highlighting the interaction between augmentation types and model performance across the datasets.

- Table D.2 breaks down the augmentation-framework ablation study when employing a multimodal early-fusion approach. It details the impact of augmentations for each event representation, enabling a detailed comparison across different modalities and datasets.

- Table D.3 focuses on the late-fusion approach in the multi-modal setting. It provides an exhaustive analysis of the augmentation effects, ensuring a comprehensive understanding of the augmentation framework's influence on various spatio-temporal representations.

These results are vital for grasping the complex dynamics of augmentation frameworks in event-based object detection. They validate the critical role of spatial augmentations in object detection tasks and highlight the supporting roles of randomized temporal, polarity, and event drop augmentations. Together, these insights contribute to the broader goal of enhancing event-based models for improved accuracy and generalization.

Table D.1: Results of the augmentation-framework ablation study using the event-based vision modality, detailing the results presented in Table 6.15 for each event representation. The effects of each augmentation configuration are demonstrated for each spatio-temporal representation explored in this work.

Representation	**Augmentation Type**				**MEVDT** (ΔT = 500 ms)			**PKU-DDD17-CAR**		
	Spatial	Temporal	Polarity	Drop	mAP (%)	mAP_{75} (%)	mAP_{50} (%)	mAP (%)	mAP_{75} (%)	mAP_{50} (%)
Timestamp Image	Baseline				80.8	94.5	97.5	21.7	19.3	44.3
				✓	82.5	94.8	97.7	21.7	17.9	45.2
			✓		82.3	93.8	97.7	21.3	17.8	45.1
		✓			83.2	94.6	97.5	21.8	18.3	45.0
		✓	✓		82.1	93.8	97.7	21.6	18.2	44.5
		✓	✓	✓	81.5	94.7	98.4	21.5	18.5	43.9
	✓				84.8	94.6	97.7	24.4	**22.7**	49.0
	✓	✓	✓		84.7	94.9	98.8	23.2	19.7	48.8
	✓	✓	✓	✓	83.2	94.7	97.5	23.1	20.3	49.4
Timestamp Image & Count	Baseline				80.6	93.2	97.0	21.6	18.2	46.3
				✓	81.6	94.8	97.6	22.0	17.7	46.4
			✓		81.5	93.8	96.8	22.4	20.5	45.9
		✓			82.9	95.3	98.6	22.1	18.8	46.0
		✓	✓		82.5	94.7	97.8	21.8	18.8	45.7
		✓	✓	✓	81.6	95.7	98.6	22.5	19.1	46.9
	✓				84.1	93.8	96.9	24.3	21.7	49.7
	✓	✓	✓		**85.1**	**95.8**	98.7	23.8	20.9	48.7
	✓	✓	✓	✓	84.6	93.9	98.7	**24.7**	22.7	**51.1**
CSTR (mean Ts only)	Baseline				81.5	94.5	97.7	21.4	18.1	44.5
				✓	81.8	92.5	97.5	21.4	17.8	44.5
			✓		79.4	91.4	96.3	20.9	17.5	44.1
		✓			82.1	94.6	97.8	22.3	20.4	44.8
		✓	✓		81.6	93.6	98.4	21.3	18.0	44.5
		✓	✓	✓	81.1	93.6	97.6	21.2	17.5	44.6
	✓				83.1	93.1	97.5	23.2	18.8	48.5
	✓	✓	✓		84.3	94.5	98.7	23.6	20.3	49.0
	✓	✓	✓	✓	84.2	93.4	98.7	23.7	20.4	48.7
CSTR (mean Ts & Count)	Baseline				80.5	94.5	97.5	22.0	17.9	45.7
				✓	81.3	94.2	97.6	23.8	21.3	47.9
			✓		82.2	93.5	97.5	21.8	18.7	45.8
		✓			83.4	94.5	98.2	21.9	18.1	46.3
		✓	✓		81.3	93.5	97.3	22.0	19.2	45.6
		✓	✓	✓	81.7	94.5	98.4	23.2	20.1	47.2
	✓				84.1	93.5	96.9	24.1	21.1	50.4
	✓	✓	✓		84.9	94.3	98.6	24.4	22.0	50.6
	✓	✓	✓	✓	84.9	94.5	**98.8**	24.2	21.9	49.9
Average	Baseline				80.8	94.2	97.4	21.7	18.4	45.2
				✓	81.8	94.1	97.6	22.2	18.6	46.0
			✓		81.4	93.1	97.1	21.6	18.6	45.2
		✓			82.9	94.8	98.0	22.0	18.9	45.5
		✓	✓		81.9	93.9	97.8	21.7	18.6	45.1
		✓	✓	✓	81.5	94.6	98.2	22.1	18.8	45.6
	✓				84.0	93.8	97.3	**24.0**	21.1	49.4
	✓	✓	✓		**84.8**	**94.9**	**98.7**	23.7	20.7	49.3
	✓	✓	✓	✓	84.2	94.1	98.4	23.9	**21.3**	**49.8**

The best results per dataset and metric are highlighted in **bold**.

Table D.2: Results of the augmentation-framework ablation study using the multi-modal early-fusion approach, detailing the results presented in Table 6.15 for each event representation. The effects of each augmentation configuration are demonstrated for each spatio-temporal representation explored in this work.

Representation	**Augmentation Type**				**MEVDT** (ΔT = 500 ms)			**PKU-DDD17-CAR**		
	Spatial	Temporal	Polarity	Drop	mAP (%)	mAP_{75} (%)	mAP_{50} (%)	mAP (%)	mAP_{75} (%)	mAP_{50} (%)
Timestamp Image	Baseline				86.8	96.9	98.9	45.2	41.9	87.3
				✓	85.9	97.0	98.9	43.7	37.9	87.1
			✓		84.7	96.8	98.9	44.6	40.3	87.8
		✓			87.2	96.9	98.9	44.6	40.5	87.6
		✓	✓		87.2	97.0	98.9	43.4	37.5	88.1
		✓	✓	✓	87.2	97.8	98.9	44.4	38.0	87.9
	✓				87.5	97.0	99.0	47.3	44.2	91.1
	✓	✓	✓		86.4	96.8	98.8	47.1	41.7	91.2
	✓	✓	✓	✓	88.0	96.9	98.9	45.5	41.0	89.0
Timestamp Image & Count	Baseline				86.6	97.0	99.0	44.6	42.2	87.3
				✓	85.9	97.0	99.0	42.8	36.4	87.3
			✓		85.7	97.0	99.0	42.7	36.4	86.4
		✓			87.0	97.0	98.9	44.1	37.9	88.1
		✓	✓		86.7	96.9	98.9	44.2	39.7	87.2
		✓	✓	✓	86.2	97.0	98.9	44.7	40.7	87.2
	✓				88.8	96.9	99.0	48.2	**46.1**	91.2
	✓	✓	✓		86.4	96.9	99.0	47.5	44.0	91.1
	✓	✓	✓	✓	87.1	97.0	98.9	45.4	39.8	89.5
CSTR (mean Ts only)	Baseline				87.7	97.0	**99.0**	44.5	39.5	87.4
				✓	86.8	96.9	98.9	45.2	42.0	88.5
			✓		86.0	97.0	99.0	44.6	41.1	87.5
		✓			86.9	96.9	98.9	45.6	42.7	87.9
		✓	✓		87.1	97.0	98.9	44.6	39.2	87.9
		✓	✓	✓	85.9	96.9	98.9	45.9	44.7	85.7
	✓				86.3	96.9	98.9	**48.4**	45.7	**91.9**
	✓	✓	✓		87.1	96.8	98.8	47.5	44.0	90.2
	✓	✓	✓	✓	**89.4**	96.9	98.8	47.2	44.1	90.6
CSTR (mean Ts & Count)	Baseline				86.5	97.0	99.0	44.4	37.7	87.9
				✓	87.2	96.9	98.9	44.8	41.3	87.3
			✓		85.0	96.9	98.9	44.8	40.2	87.9
		✓			83.8	96.9	98.9	44.5	40.4	86.8
		✓	✓		86.0	96.9	98.9	45.1	41.3	86.9
		✓	✓	✓	87.4	97.0	99.0	45.0	41.0	88.3
	✓				87.3	97.0	99.0	46.7	41.7	90.6
	✓	✓	✓		87.0	97.0	99.0	43.3	34.8	88.9
	✓	✓	✓	✓	87.1	**97.9**	99.0	46.5	42.8	89.8
Average	Baseline				86.9	96.9	**99.0**	44.7	40.3	87.5
				✓	86.5	96.9	98.9	44.2	39.4	87.6
			✓		85.4	96.9	98.9	44.1	39.5	87.4
		✓			86.2	96.9	98.9	44.7	40.4	87.6
		✓	✓		86.8	97.0	98.9	44.3	39.4	87.5
		✓	✓	✓	86.7	**97.2**	98.9	45.0	41.1	87.3
	✓				87.5	96.9	98.9	**47.6**	**44.4**	**91.2**
	✓	✓	✓		86.7	96.9	98.9	46.3	41.1	90.4
	✓	✓	✓	✓	**87.9**	97.2	98.9	46.2	41.9	89.8

The best results per dataset and metric are highlighted in **bold**.

Table D.3: Results of the augmentation-framework ablation study using the multi-modal late-fusion approach, detailing the results presented in Table 6.15 for each event representation. The effects of each augmentation configuration are demonstrated for each spatio-temporal representation explored in this work.

Representation	**Augmentation Type**				**MEVDT** (ΔT = 500 ms)			**PKU-DDD17-CAR**		
	Spatial	Temporal	Polarity	Drop	mAP (%)	mAP_{75} (%)	mAP_{50} (%)	mAP (%)	mAP_{75} (%)	mAP_{50} (%)
Timestamp Image	Baseline				84.6	96.9	**98.9**	43.4	40.4	84.1
				✓	83.4	96.5	98.5	43.2	40.0	82.9
			✓		84.8	96.6	98.8	44.3	42.4	84.6
		✓			82.1	96.5	98.6	45.0	43.4	84.1
		✓	✓		84.2	96.6	98.7	44.2	41.9	84.8
		✓	✓	✓	81.9	97.5	98.7	43.8	42.4	82.3
	✓				87.6	97.7	98.7	44.8	43.3	83.9
	✓	✓	✓		86.9	96.8	98.7	45.0	41.5	85.9
	✓	✓	✓	✓	87.5	97.6	98.6	46.9	46.1	87.1
Timestamp Image & Count	Baseline				85.2	96.8	98.8	43.6	41.3	84.3
				✓	86.1	97.8	98.9	43.8	41.5	83.9
			✓		85.2	96.7	98.8	43.2	39.4	85.3
		✓			83.4	96.9	98.9	43.4	39.8	83.9
		✓	✓		83.6	97.8	98.8	44.1	42.6	84.4
		✓	✓	✓	84.3	97.6	98.7	44.4	42.4	84.0
	✓				87.3	97.8	98.8	45.1	44.6	84.2
	✓	✓	✓		86.9	96.7	98.6	**48.4**	**49.4**	**88.0**
	✓	✓	✓	✓	**88.2**	97.7	98.8	45.2	42.6	85.8
CSTR (mean Ts only)	Baseline				85.1	96.8	98.9	43.8	41.8	85.1
				✓	83.7	96.5	98.7	44.3	41.2	84.4
			✓		83.7	96.4	98.7	42.7	39.4	83.0
		✓			84.3	**97.8**	98.9	43.0	41.5	82.9
		✓	✓		84.9	97.5	98.8	43.4	42.5	83.6
		✓	✓	✓	83.6	96.9	98.8	44.5	41.8	84.0
	✓				85.7	96.7	98.9	47.3	47.5	87.7
	✓	✓	✓		86.9	96.4	98.7	45.8	44.0	85.9
	✓	✓	✓	✓	86.2	96.4	98.8	45.5	44.2	84.5
CSTR (mean Ts & Count)	Baseline				84.7	96.8	98.9	43.6	41.0	84.3
				✓	85.3	97.6	98.8	44.4	43.4	83.3
			✓		83.7	96.1	98.7	44.7	42.1	85.8
		✓			85.7	97.7	98.8	43.6	40.7	83.6
		✓	✓		82.3	96.5	98.7	43.6	41.0	83.4
		✓	✓	✓	83.4	96.6	98.6	44.1	42.9	82.9
	✓				85.5	96.6	98.5	47.2	46.9	87.1
	✓	✓	✓		87.1	96.5	98.7	44.6	42.4	85.4
	✓	✓	✓	✓	87.7	97.5	98.7	45.2	41.1	86.3
Average	Baseline				84.9	96.8	**98.9**	43.6	41.1	84.4
				✓	84.6	97.1	98.7	43.9	41.5	83.6
			✓		84.4	96.4	98.7	43.7	40.8	84.7
		✓			83.9	97.2	98.8	43.8	41.4	83.6
		✓	✓		83.8	97.1	98.7	43.8	42.0	84.1
		✓	✓	✓	83.3	97.2	98.7	44.2	42.4	83.3
	✓				86.5	97.2	98.7	**46.1**	**45.6**	85.7
	✓	✓	✓		86.9	96.6	98.7	46.0	44.3	**86.3**
	✓	✓	✓	✓	**87.4**	**97.3**	98.7	45.7	43.5	85.9

The best results per dataset and metric are highlighted in **bold**.

BIBLIOGRAPHY

[1] Mohamed Aladem, Sumanth Chennupati, Zaid El-Shair, and Samir A. Rawashdeh. A comparative study of different cnn encoders for monocular depth prediction. In *2019 IEEE National Aerospace and Electronics Conference (NAECON)*, pages 328–331. IEEE, 2019.

[2] Mohamed Aladem and Samir A Rawashdeh. A combined vision-based multiple object tracking and visual odometry system. *IEEE sensors journal*, 19(23):11714–11720, 2019.

[3] Inigo Alonso and Ana C Murillo. Ev-segnet: Semantic segmentation for event-based cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[4] Fouzia Altaf, Syed MS Islam, Naveed Akhtar, and Naeem Khalid Janjua. Going deep in medical image analysis: concepts, methods, challenges, and future directions. *IEEE Access*, 7:99540–99572, 2019.

[5] Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, et al. A low power, fully event-based gesture recognition system. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7243–7252, 2017.

[6] Irfan Baftiu, Arbnor Pajaziti, and Ka C Cheok. Multi-mode surround view for adas vehicles. In *2016 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*, pages 190–193. IEEE, 2016.

[7] Fariborz Baghaei Naeini, Dimitrios Makris, Dongming Gan, and Yahya Zweiri. Dynamic-vision-based force measurements using convolutional recurrent neural networks. *Sensors*, 20(16):4469, 2020.

[8] WeiJie Bai, Yunhua Chen, Ren Feng, and Yuliang Zheng. Accurate and efficient frame-based event representation for aer object recognition. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE, 2022.

[9] Raymond Baldwin, Ruixu Liu, Mohammed Mutlaq Almatrafi, Vijayan K Asari, and Keigo Hirakawa. Time-ordered recent event (tore) volumes for event cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[10] Koyel Banerjee, Dominik Notz, Johannes Windelen, Sumanth Gavarraju, and Mingkang He. Online camera lidar fusion and object detection on hybrid data for autonomous driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1632–1638. IEEE, 2018.

[11] Francisco Barranco, Cornelia Fermuller, and Eduardo Ros. Real-time clustering and multi-target tracking using event-based sensors. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5764–5769. IEEE, 2018.

[12] Chiara Bartolozzi, Giacomo Indiveri, and Elisa Donati. Embodied neuromorphic intelligence. *Nature communications*, 13(1):1–14, 2022.

[13] Rodrigo Benenson, Stéphane Petti, Thierry Fraichard, and Michel Parent. Towards urban driverless vehicles. *International journal of vehicle autonomous systems*, 6(1-2):4–23, 2008.

[14] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[15] Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatze, and Yiannis Andreopoulos. Graph-based object classification for neuromorphic vision sensing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 491–501, 2019.

[16] Yin Bi, Aaron Chadha, Alhabib Abbas, Eirina Bourtsoulatze, and Yiannis Andreopoulos. Graph-based spatio-temporal feature learning for neuromorphic vision sensing. *IEEE Transactions on Image Processing*, 29:9084–9098, 2020.

[17] Jonathan Binas, Daniel Neil, Shih-Chii Liu, and Tobi Delbruck. Ddd17: End-to-end davis driving dataset. *arXiv preprint arXiv:1711.01458*, 2017.

[18] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[19] Tobias Bolten, Regina Pohle-Frohlich, and Klaus D Tonnies. Dvs-outlab: A neuromorphic event-based long time monitoring dataset for real-world outdoor scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1348–1357, 2021.

[20] Andrea Bonci, Pangcheng David Cen Cheng, Marina Indri, Giacomo Nabissi, and Fiorella Sibona. Human-robot perception in industrial environments: A survey. *Sensors*, 21(5):1571, 2021.

[21] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240$\times$ 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits*, 49(10):2333–2341, 2014.

[22] Wilhelm Burger and Bir Bhanu. Estimating 3d egomotion from perspective image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(11):1040–1058, 1990.

[23] Stephen C Cain, Majeed M Hayat, and Ernest E Armstrong. Projection-based image registration in the presence of fixed-pattern noise. *IEEE transactions on image processing*, 10(12):1860–1872, 2001.

[24] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. Asynchronous convolutional networks for object detection in neuromorphic cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[25] John F Canny. A variational approach to edge detection. In *AAAI*, volume 1983, pages 54–58, 1983.

[26] Hu Cao, Guang Chen, Jiahao Xia, Genghang Zhuang, and Alois Knoll. Fusion-based feature attention gate component for vehicle detection based on event camera. *IEEE Sensors Journal*, 21(21):24540–24548, 2021.

[27] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[28] Andrea Censi and Davide Scaramuzza. Low-latency event-based visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 703–710. IEEE, 2014.

[29] Guang Chen, Hu Cao, Jorg Conradt, Huajin Tang, Florian Rohrbein, and Alois Knoll. Event-based neuromorphic vision for autonomous driving: A paradigm shift for bio-inspired visual sensing and perception. *IEEE Signal Processing Magazine*, 37(4):34–49, 2020.

[30] Guo Chen, Li Li, Weiqi Jin, and Shuo Li. High-dynamic range, night vision, image-fusion algorithm based on a decomposition convolution neural network. *IEEE Access*, 7:169762–169772, 2019.

[31] Haosheng Chen, Qiangqiang Wu, Yanjie Liang, Xinbo Gao, and Hanzi Wang. Asynchronous tracking-by-detection on adaptive time surfaces for event-based object tracking. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 473–481, 2019.

[32] Nicholas FY Chen. Pseudo-labels for supervised learning on dynamic vision sensor data, applied to object detection under ego-motion. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 644–653, 2018.

[33] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):569–582, 2014.

[34] Aaron Cofield, Zaid El-Shair, and Samir A. Rawashdeh. A humanoid robot object perception approach using depth images. In *2019 IEEE National Aerospace and Electronics Conference (NAECON)*, pages 437–442. IEEE, 2019.

[35] Giorgio Cruciata, Liliana Lo Presti, and Marco La Cascia. Iterative multiple bounding-box refinements for visual tracking. *Journal of Imaging*, 8(3):61, 2022.

[36] Per-Erik Danielsson. Euclidean distance mapping. *Computer Graphics and image processing*, 14(3):227–248, 1980.

[37] Gabriel Silva de Oliveira, José Marcato Junior, Caio Polidoro, Lucas Prado Osco, Henrique Siqueira, Lucas Rodrigues, Liana Jank, Sanzio Barrios, Cacilda Valle, Rosângela Simeão, et al. Convolutional neural networks to estimate dry matter yield in a guineagrass breeding program using uav remote sensing. *Sensors*, 21(12):3971, 2021.

[38] Patrick Dendorfer, Aljosa Osep, Anton Milan, Konrad Schindler, Daniel Cremers, Ian Reid, Stefan Roth, and Laura Leal-Taixé. Motchallenge: A benchmark for single-camera multiple target tracking. *International Journal of Computer Vision*, 129:845–881, 2021.

[39] Patrick Dendorfer, Hamid Rezatofighi, Anton Milan, Javen Shi, Daniel Cremers, Ian Reid, Stefan Roth, Konrad Schindler, and Laura Leal-Taixé. Mot20: A benchmark for multi object tracking in crowded scenes. *arXiv preprint arXiv:2003.09003*, 2020.

[40] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[41] Barga Deori and Dalton Meitei Thounaojam. A survey on moving object tracking in video. *International Journal on Information Theory (IJIT)*, 3(3):31–46, 2014.

[42] Junfei Dong, Runhao Jiang, Rong Xiao, Rui Yan, and Huajin Tang. Event stream learning using spatio-temporal event surface. *Neural Networks*, 154:543–559, 2022.

[43] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[44] Zaid El Shair and Samir Rawashdeh. High-temporal-resolution event-based vehicle detection and tracking. *Optical Engineering*, 62(3):031209–031209, 2023.

[45] Zaid El Shair and Samir A. Rawashdeh. High-temporal-resolution object detection and tracking using images and events. *Journal of Imaging*, 8(8):210, 2022.

[46] Zaid A. El Shair, Ali Hassani, and Samir A. Rawashdeh. CSTR: A Compact Spatio-Temporal Representation for Event-Based Vision. *IEEE Access*, 11:102899–102916, 2023.

[47] Zaid A. El-Shair and Samir A. Rawashdeh. Design of an object sorting system using a vision-guided robotic arm. In *American Society for Engineering Education North-Central Section Conference*, 2019.

[48] Zaid A. El-Shair, Luis A. Sánchez-Pérez, and Samir A. Rawashdeh. Comparative study of machine learning algorithms using a breast cancer dataset. In *2020 IEEE International Conference on Electro Information Technology (EIT)*, pages 500–508. IEEE, 2020.

[49] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[50] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[51] Davide Falanga, Suseong Kim, and Davide Scaramuzza. How fast is too fast? the role of perception latency in high-speed sense and avoid. *IEEE Robotics and Automation Letters*, 4(2):1884–1891, 2019.

[52] Davide Falanga, Kevin Kleber, and Davide Scaramuzza. Dynamic obstacle avoidance for quadrotors with event cameras. *Science Robotics*, 5(40):eaaz9712, 2020.

[53] Wei Fang, Zhaofei Yu, Yanqi Chen, Timothée Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating learnable membrane time constant to enhance learning of spiking neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2661–2671, 2021.

[54] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020.

[55] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020.

[56] Guillermo Gallego, Jon EA Lund, Elias Mueggler, Henri Rebecq, Tobi Delbruck, and Davide Scaramuzza. Event-based, 6-dof camera tracking from photometric depth maps. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2402–2412, 2017.

[57] Daniel Gehrig, Antonio Loquercio, Konstantinos G Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5633–5643, 2019.

[58] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. Eklt: Asynchronous photometric feature tracking using events and frames. *International Journal of Computer Vision*, 128(3):601–618, 2020.

[59] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. Dsec: A stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters*, 6(3):4947–4954, 2021.

[60] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[61] Michael Gerstmair, Martin Gschwandtner, Rainer Findenig, Alexander Melzer, and Mario Huemer. Lego radar train—an educational workshop on radar-based advanced driver assistance systems. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 1981–1985. IEEE, 2021.

[62] Rohan Ghosh, Abhishek Mishra, Garrick Orchard, and Nitish V Thakor. Real-time object recognition and orientation estimation using an event-based camera and cnn. In *2014 IEEE Biomedical Circuits and Systems Conference (BioCAS) Proceedings*, pages 544–547. IEEE, 2014.

[63] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[64] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[65] Fuqiang Gu, Weicong Sng, Xuke Hu, and Fangwen Yu. Eventdrop: Data augmentation for event-based learning. In *30th International Joint Conference on Artificial Intelligence, IJCAI 202*, 2021.

[66] Julio Guillen-Garcia, Daniel Palacios-Alonso, Enrique Cabello, and Cristina Conde. Unsupervised adaptive multi-object tracking-by-clustering algorithm with a bio-inspired system. *IEEE Access*, 10:24895–24908, 2022.

[67] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27–48, 2016.

[68] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4338–4364, 2020.

[69] Yazan Hamzeh, Zaid El-Shair, and Samir A. Rawashdeh. Effect of adherent rain on vision-based object detection algorithms. *SAE International Journal of Advances and Current Practices in Mobility*, 2(2020-01-0104):3051–3059, 2020.

[70] Yazan Hamzeh, Zaid A. El-Shair, Abdallah Chehade, and Samir A. Rawashdeh. Dynamic adherent raindrop simulator for automotive vision systems. *IEEE Access*, 9:114808–114820, 2021.

[71] Ali Hassani, Zaid El Shair, Rafi Ud Duala Refat, and Hafiz Malik. Distilling facial knowledge with teacher-tasks: Semantic-segmentation-features for pose-invariant face-recognition. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 741–745. IEEE, 2022.

[72] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[73] Dirk Hertel. Extended use of iso 15739 incremental signal-to-noise ratio as reliability criterion for multiple-slope wide dynamic range image capture. In *Image Quality and System Performance VI*, volume 7242, pages 85–94. SPIE, 2009.

[74] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4507–4515, 2017.

[75] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.

[76] Jun-Wei Hsieh, Shih-Hao Yu, Yung-Sheng Chen, and Wen-Fong Hu. Automatic traffic surveillance system for vehicle tracking and classification. *IEEE Transactions on intelligent transportation systems*, 7(2):175–187, 2006.

[77] Craig Iaboni, Himanshu Patel, Deepan Lobo, Ji-Won Choi, and Pramod Abichandani. Event camera based real-time detection and tracking of indoor ground robots. *IEEE Access*, 9:166588–166602, 2021.

[78] Massimiliano Iacono, Stefan Weber, Arren Glover, and Chiara Bartolozzi. Towards event-driven object detection with off-the-shelf deep learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9. IEEE, 2018.

[79] Paul Jaccard. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50, 1912.

[80] Zhe Jiang, Yu Zhang, Dongqing Zou, Jimmy Ren, Jiancheng Lv, and Yebin Liu. Learning event-based motion deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3320–3329, 2020.

[81] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. A survey of deep learning-based object detection. *IEEE access*, 7:128837–128868, 2019.

[82] Arne Hoffhues Jonathon Luiten. Trackeval. https://github.com/JonathonLuiten/TrackEval, 2020.

[83] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[84] Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew Davison. Simultaneous mosaicing and tracking with an event camera. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[85] Il-Hwan Kim, Jae-Hwan Bong, Jooyoung Park, and Shinsuk Park. Prediction of driver's intention of lane change by augmenting sensor information using machine learning techniques. *Sensors*, 17(6):1350, 2017.

[86] Junho Kim, Jaehyeok Bae, Gangin Park, Dongsu Zhang, and Young Min Kim. N-imagenet: Towards robust, fine-grained object recognition with event cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2146–2156, 2021.

[87] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations (ICLR)*, 2015.

[88] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[89] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[90] Seong Kyung Kwon, Eugin Hyun, Jin-Hee Lee, Jonghun Lee, and Sang Hyuk Son. A low-complexity scheme for partially occluded pedestrian detection using lidar-radar sensor fusion. In *2016 IEEE 22nd international conference on embedded and real-time computing systems and applications (RTCSA)*, pages 104–104. IEEE, 2016.

[91] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E Shi, and Ryad B Benosman. Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1346–1359, 2016.

[92] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[93] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in neuroscience*, 11:309, 2017.

[94] Hongmin Li and Luping Shi. Robust event-based object tracking combining correlation filter and cnn representation. *Frontiers in neurorobotics*, 13:82, 2019.

[95] Jianing Li, Siwei Dong, Zhaofei Yu, Yonghong Tian, and Tiejun Huang. Event-based vision enhanced: A joint detection framework in autonomous driving. In *2019 ieee international conference on multimedia and expo (icme)*, pages 1396–1401. IEEE, 2019.

[96] Jianing Li, Jia Li, Lin Zhu, Xijie Xiang, Tiejun Huang, and Yonghong Tian. Asynchronous spatio-temporal memory network for continuous event-based object detection. *IEEE Transactions on Image Processing*, 31:2975–2987, 2022.

[97] Xirong Li, Tiberio Uricchio, Lamberto Ballan, Marco Bertini, Cees GM Snoek, and Alberto Del Bimbo. Socializing the semantic gap: A comparative survey on image tag assignment, refinement, and retrieval. *ACM Computing Surveys (CSUR)*, 49(1):1–39, 2016.

[98] Yuhang Li, Youngeun Kim, Hyoungseob Park, Tamar Geller, and Priyadarshini Panda. Neuromorphic data augmentation for training spiking neural networks. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 631–649. Springer, 2022.

[99] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7345–7353, 2019.

[100] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128 x 128 120db 30mw asynchronous vision sensor that responds to relative intensity change. In *2006 IEEE International Solid State Circuits Conference-Digest of Technical Papers*, pages 2060–2069. IEEE, 2006.

[101] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128$\times$128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576, 2008.

[102] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2018.

[103] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[104] Hongjie Liu, Diederik Paul Moeys, Gautham Das, Daniel Neil, Shih-Chii Liu, and Tobi Delbrück. Combined frame-and event-based detection and tracking. In *2016 IEEE International Symposium on Circuits and systems (ISCAS)*, pages 2511–2514. IEEE, 2016.

[105] Mengyun Liu, Na Qi, Yunhui Shi, and Baocai Yin. An attention fusion network for event-based vehicle object detection. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3363–3367. IEEE, 2021.

[106] Min Liu and Tobi Delbruck. Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. In *British Machine Vision Conference (BMVC) 2018*. BMVC, 2018.

[107] Shih-Chii Liu, Tobi Delbruck, Giacomo Indiveri, Adrian Whatley, and Rodney Douglas. *Event-based neuromorphic systems*. John Wiley & Sons, 2014.

[108] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[109] Yuhao Liu, Felipe Gutierrez-Barragan, Atul Ingle, Mohit Gupta, and Andreas Velten. Single-photon camera guided extreme dynamic range imaging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1575–1585, 2022.

[110] Weixin Lu, Yao Zhou, Guowei Wan, Shenhua Hou, and Shiyu Song. L3-net: Towards learning based lidar localization for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6389–6398, 2019.

[111] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, pages 1–31, 2020.

[112] Misha Mahowald. Vlsi analogs of neuronal visual processing: a synthesis of form and function. 1992.

[113] Jiayuan Mao, Tete Xiao, Yuning Jiang, and Zhimin Cao. What can help pedestrian detection? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3127–3136, 2017.

[114] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5419–5427, 2018.

[115] Enrique Marti, Miguel Angel De Miguel, Fernando Garcia, and Joshue Perez. A review of sensor technologies for perception in automated driving. *IEEE Intelligent Transportation Systems Magazine*, 11(4):94–108, 2019.

[116] Ishaan Mehta, Mingliang Tang, and Timothy D Barfoot. Gradient-based auto-exposure control applied to a self-driving car. In *2020 17th Conference on Computer and Robot Vision (CRV)*, pages 166–173. IEEE, 2020.

[117] Qingyan Meng, Mingqing Xiao, Shen Yan, Yisen Wang, Zhouchen Lin, and Zhi-Quan Luo. Training high-performance low-latency spiking neural networks by differentiation on spike representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12444–12453, 2022.

[118] Christopher A Metzler, Hayato Ikoma, Yifan Peng, and Gordon Wetzstein. Deep optics for single-shot high-dynamic-range imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1375–1385, 2020.

[119] Anton Mitrokhin, Cornelia Fermüller, Chethan Parameshwara, and Yiannis Aloimonos. Event-based moving object detection and tracking. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9. IEEE, 2018.

[120] Erick Moen, Dylan Bannon, Takamasa Kudo, William Graf, Markus Covert, and David Van Valen. Deep learning for cellular image analysis. *Nature methods*, 16(12):1233–1246, 2019.

[121] Anindya Mondal, Jhony H Giraldo, Thierry Bouwmans, Ananda S Chowdhury, et al. Moving object detection for event-based vision using graph spectral clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 876–884, 2021.

[122] Elias Mueggler, Basil Huber, and Davide Scaramuzza. Event-based, 6-dof pose tracking for high-speed maneuvers. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2761–2768. IEEE, 2014.

[123] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *The International Journal of Robotics Research*, 36(2):142–149, 2017.

[124] Fariborz Baghaei Naeini, Sanket Kachole, Rajkumar Muthusamy, Dimitrios Makris, and Yahya Zweiri. Event augmentation for contact force measurements. *IEEE Access*, 10:123651–123660, 2022.

[125] National Transportation Safety Board (NTSB). Collision between vehicle controlled by developmental automated driving system and pedestrian. https://www.ntsb.gov/investigations/Pages/HWY18MH010.aspx, March 2018. [Accessed 18 March 2022].

[126] Isaac Ogunrinde and Shonda Bernadin. A review of the impacts of defogging on deep learning-based object detectors in self-driving cars. *SoutheastCon 2021*, pages 01–08, 2021.

[127] Isaac Ogunrinde and Shonda Bernadin. Deep camera–radar fusion with an attention framework for autonomous vehicle vision in foggy weather conditions. *Sensors*, 23(14), 2023.

[128] Alan Ohnsman. LiDAR Maker Velodyne 'Baffled' By Self-Driving Uber's Failure To Avoid Pedestrian — forbes.com. https://www.forbes.com/sites/alanohnsman/2018/03/23/lidar-maker-velodyne-baffled-by-self-driving-ubers-failure-to-avoid-pedestrian, March 2018. [Accessed 21 March 2022].

[129] Emmanuel Onzon, Fahim Mannan, and Felix Heide. Neural auto-exposure for high-dynamic range object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7710–7720, 2021.

[130] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:437, 2015.

[131] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[132] Liyuan Pan, Cedric Scheerlinck, Xin Yu, Richard Hartley, Miaomiao Liu, and Yuchao Dai. Bringing a blurry frame alive at high frame-rate with an event camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6820–6829, 2019.

[133] Federico Paredes-Vallés, Kirk YW Scheper, and Guido CHE De Croon. Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception. *IEEE transactions on pattern analysis and machine intelligence*, 42(8):2051–2064, 2019.

[134] Paul KJ Park, Baek Hwan Cho, Jin Man Park, Kyoobin Lee, Ha Young Kim, Hyo Ah Kang, Hyun Goo Lee, Jooyeon Woo, Yohan Roh, Won Jo Lee, et al. Performance improvement of deep learning based gesture recognition using spatiotemporal demosaicing technique. In

2016 IEEE International Conference on Image Processing (ICIP), pages 1624–1628. IEEE, 2016.

[135] Jinlong Peng, Tao Wang, Weiyao Lin, Jian Wang, John See, Shilei Wen, and Erui Ding. Tpm: Multiple object tracking with tracklet-plane matching. *Pattern Recognition*, 107:107480, 2020.

[136] Etienne Perot, Pierre de Tournemire, Davide Nitti, Jonathan Masci, and Amos Sironi. Learning to detect objects with a 1 megapixel event camera. *Advances in Neural Information Processing Systems*, 33:16639–16652, 2020.

[137] Michael Pfeiffer and Thomas Pfeil. Deep learning with spiking neurons: Opportunities and challenges. *Frontiers in neuroscience*, 12:774, 2018.

[138] Jinan Piao and Mike McDonald. Advanced driver assistance systems from autonomous to cooperative approach. *Transport reviews*, 28(5):659–684, 2008.

[139] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds. *IEEE Journal of Solid-State Circuits*, 46(1):259–275, 2010.

[140] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds. *IEEE Journal of Solid-State Circuits*, 46(1):259–275, 2010.

[141] Christoph Posch, Teresa Serrano-Gotarredona, Bernabe Linares-Barranco, and Tobi Delbruck. Retinomorphic event-based vision sensors: bioinspired cameras with spiking output. *Proceedings of the IEEE*, 102(10):1470–1484, 2014.

[142] Basilio Pueo. High speed cameras for motion analysis in sports science. *Journal of Human Sport and Exercise*, 11(1):53–73, 2016.

[143] Bharath Ramesh, Shihao Zhang, Zhi Wei Lee, Zhi Gao, Garrick Orchard, and Cheng Xiang. Long-term object tracking with a moving event camera. In *Bmvc*, page 241, 2018.

[144] Bharath Ramesh, Shihao Zhang, Hong Yang, Andres Ussa, Matthew Ong, Garrick Orchard, and Cheng Xiang. e-tld: Event-based framework for dynamic object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3996–4006, 2020.

[145] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):121–135, 2017.

[146] Ratheesh Ravindran, Michael J Santora, and Mohsin M Jamali. Multi-object detection and tracking, based on dnn, for autonomous vehicles: A review. *IEEE Sensors Journal*, 21(5):5668–5677, 2020.

[147] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3857–3866, 2019.

[148] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE transactions on pattern analysis and machine intelligence*, 43(6):1964–1980, 2019.

[149] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[150] Haotian Ren, Wei Lu, Yun Xiao, Xiaojun Chang, Xuanhong Wang, Zhiqiang Dong, and Dingyi Fang. Graph convolutional networks in language and vision: A survey. *Knowledge-Based Systems*, page 109250, 2022.

[151] Jimmy Ren, Xiaohao Chen, Jianbo Liu, Wenxiu Sun, Jiahao Pang, Qiong Yan, Yu-Wing Tai, and Li Xu. Accurate single stage detector using recurrent rolling convolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5420–5428, 2017.

[152] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[153] Adrian Rosebrock. Simple object tracking with opencv. *PyImageSearch. Available online: https://www. pyimagesearch. com/2018/07/23/simple-object-tracking-with-opencv/(accessed on 1 October 2021)*, 2018.

[154] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[155] Simon Schaefer, Daniel Gehrig, and Davide Scaramuzza. Aegnn: Asynchronous event-based graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12371–12381, 2022.

[156] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Continuous-time intensity estimation using event cameras. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part V*, pages 308–324. Springer, 2019.

[157] Aditya Sharma. Mean Average Precision (mAP) Using the COCO Evaluator - PyImageSearch — pyimagesearch.com. https://pyimagesearch.com/2022/05/02/mean-average-precision-map-using-the-coco-evaluator/, 2022. [Accessed 27-12-2023].

[158] Guobin Shen, Dongcheng Zhao, and Yi Zeng. Eventmix: An efficient data augmentation strategy for event-based learning. *Information Sciences*, page 119170, 2023.

[159] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[160] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[161] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1731–1740, 2018.

[162] Yunjae Suh, Seungnam Choi, Masamichi Ito, Jeongseok Kim, Youngho Lee, Jongseok Seo, Heejae Jung, Dong-Hee Yeo, Seol Namgung, Jongwoo Bong, et al. A 1280× 960 dynamic vision sensor with a 4.95-μm pixel pitch and motion artifact minimization. In *2020 IEEE international symposium on circuits and systems (ISCAS)*, pages 1–5. IEEE, 2020.

[163] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[164] The Tesla Team. A tragic loss. https://www.tesla.com/blog/tragic-loss, June 2016. [Accessed 18 March 2022].

[165] David Tedaldi, Guillermo Gallego, Elias Mueggler, and Davide Scaramuzza. Feature detection and tracking with the dynamic and active-pixel vision sensor (davis). In *2016 Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, pages 1–7. IEEE, 2016.

[166] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[167] Abhishek Tomy, Anshul Paigwar, Khushdeep S Mann, Alessandro Renzaglia, and Christian Laugier. Fusing event-based and rgb camera for robust object detection in adverse conditions. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 933–939. IEEE, 2022.

[168] Jessica Van Brummelen, Marie O'Brien, Dominique Gruyer, and Homayoun Najjaran. Autonomous vehicle perception: The technology of today and tomorrow. *Transportation research part C: emerging technologies*, 89:384–406, 2018.

[169] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[170] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios. *IEEE Robotics and Automation Letters*, 3(2):994–1001, 2018.

[171] Rahee Walambe, Aboli Marathe, Ketan Kotecha, George Ghinea, et al. Lightweight object detection ensemble framework for autonomous vehicles in challenging weather conditions. *Computational Intelligence and Neuroscience*, 2021, 2021.

[172] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023.

[173] Qinyi Wang, Yexin Zhang, Junsong Yuan, and Yilong Lu. Space-time event clouds for gesture recognition: From rgb cameras to event cameras. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1826–1835. IEEE, 2019.

[174] Xinyu Wang, Chunhua Shen, Hanxi Li, and Shugong Xu. Human detection aided by deeply learned semantic masks. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(8):2663–2673, 2019.

[175] Yanxiang Wang, Xian Zhang, Yiran Shen, Bowen Du, Guangrong Zhao, Lizhen Cui, and Hongkai Wen. Event-stream representation for human gaits identification using deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3436–3449, 2021.

[176] Yu Wang, Xinke Ge, He Ma, Shouliang Qi, Guanjing Zhang, and Yudong Yao. Deep learning in medical ultrasound image analysis: a review. *IEEE Access*, 9:54310–54324, 2021.

[177] Mial E Warren. Automotive lidar technology. In *2019 Symposium on VLSI Circuits*, pages C254–C255. IEEE, 2019.

[178] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017.

[179] Bochen Xie, Yongjian Deng, Zhanpeng Shao, Hai Liu, and Youfu Li. Vmv-gcn: Volumetric multi-view based graph cnn for event stream classification. *IEEE Robotics and Automation Letters*, 7(2):1976–1983, 2022.

[180] Fan Yang, Xin Chang, Sakriani Sakti, Yang Wu, and Satoshi Nakamura. Remot: A model-agnostic refinement for multiple object tracking. *Image and Vision Computing*, 106:104091, 2021.

[181] Man Yao, Huanhuan Gao, Guangshe Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, and Guoqi Li. Temporal-wise attention spiking neural networks for event streams classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10221–10230, 2021.

[182] Özgün Yılmaz, Camille Simon-Chane, and Aymeric Histace. Evaluation of event-based corner detectors. *Journal of Imaging*, 7(2):25, 2021.

[183] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412, 2018.

[184] Feihu Zhang, Daniel Clarke, and Alois Knoll. Vehicle detection based on lidar and camera fusion. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1620–1625. IEEE, 2014.

[185] Jiqing Zhang, Xin Yang, Yingkai Fu, Xiaopeng Wei, Baocai Yin, and Bo Dong. Object tracking by jointly exploiting frame and event domain. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13043–13052, 2021.

[186] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, 14(2):119–135, 2017.

[187] Jiang Zhao, Shilong Ji, Zhihao Cai, Yiwen Zeng, and Yingxun Wang. Moving object detection and tracking by event frame from neuromorphic vision sensors. *Biomimetics*, 7(1):31, 2022.

[188] Linyu Zheng, Ming Tang, Yingying Chen, Guibo Zhu, Jinqiao Wang, and Hanqing Lu. Improving multiple object tracking with single object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2453–2462, 2021.

[189] Chu Zhou, Minggui Teng, Jin Han, Jinxiu Liang, Chao Xu, Gang Cao, and Boxin Shi. Deblurring low-light images with events. *International Journal of Computer Vision*, 131(5):1284–1298, 2023.

[190] Alex Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.

[191] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 989–997, 2019.

[192] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based optical flow using motion compensation. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.