Moving Objects Detection and Tracking using Hybrid Event-based and Frame-based Vision for Autonomous Driving

Haixin Sun

COLLEGE SCIENCES
DOCTORAL DE L'INGENIERIE
PAYS DE LA LOIRE ET DES SYSTEMES

CENTRALE NANTES

MEMOIRE DE DOCTORAT DE

L'ECOLE CENTRALE DE NANTES

ECOLE DOCTORALE N° 602
Sciences de l'Ingénierie et des Systèmes
Spécialité : *Robotique*

Par
Haixin SUN

Moving Objects Detection and Tracking using Hybrid Event-based and Frame-based Vision for Autonomous Driving

Projet de recherche doctoral présenté et soutenu à Nantes, le 15/06/2023
Unité de recherche : UMR 6004 Laboratoire des Sciences du Numérique de Nantes (LS2N)

Rapporteurs avant soutenance :

Pascal VASSEUR Professeur des universités, Université de Picardie Jules Verne
Rémi BOUTTEAU Professeur des universités, Université de Rouen Normandie

Composition du Jury :

Présidente : Myriam SERVIERES Professeure des universités, École Centrale de Nantes
Examinateurs : Julien MOREAU Maître de conférences, Université de technologie de Compiègne
Pascal VASSEUR Professeur des universités, Université de Picardie Jules Verne
Rémi BOUTTEAU Professeur des universités, Université de Rouen Normandie
Directeur de recherches doctorales : Vincent FREMONT Professeur des universités, École Centrale de Nantes

ACKNOWLEDGEMENT

My time as a doctoral candidate has finally come to an end. I can still vividly remember when I found out I got admitted to this GEC-CSC (Groupe Ecole Centrale - Chinese Scholarship Council) program. After months of preparation and interviews with different professors and committee panels from GEC, I felt relieved, but at the same time very thrilled, when I finally got the admission letter from CSC. I was not wrong in being excited since a new journey was about to begin.

Looking back on my three years of Ph.D. study, I would say it was a rather thorny path full of obstacles and uncertainties. It was not as pleasant and fruitful as expected, especially initially. The difficulties lay in that I did not know much about event cameras from my previous study experience. Fortunately, I received tremendous help from numerous people. My supervisor Professor Vincent FREMONT are the first people to be addressed.

Many thanks to Prof. Vincent FREMONT for his patient guidance and devoted dedication. His helpful directions and warm encouragement led me through the whole research process. The countless meeting we spent in his office exchanging thoughts on research directions and paper writings are etched into my mind.

Besides my supervisors, I would also like to express my gratitude to my CSI members, Prof. Myriam Servières and Prof. Franck DAVOINE. They have proposed enlightening questions and valuable suggestions during my CSI meetings, which helped me dive further into my research topic.

I also want to express many thanks to Minh Quan Dao. He provides me with much help regarding the research of deep learning. Without his help, I can hardly finish my thesis.

I would also like to thank all my friends and colleagues for their support and help. Especially those I met in Nantes, such as Dr. LIU Shiyu and Dr. CHEN Songming. There were many times when I was frustrated with the work or full of negative emotions; it is through consulting and talking with them that helped me relieve and regain peace in my mind.

Heartfelt gratitude goes to my parents, who provided me with an excellent environment

in which I was raised. They were attentive to my need and decisions and spared every effort for my physical and mental well-being. Their dedication, love, and trust provide me with the soil where I can thrive.

Last but not least, I would express my appreciation to my wife, who has been caring and supportive since the second I decided to pursue my Ph.D. degree in France. I can not imagine how I would pass the last four years if she were not around.

List of Publications

i. H. Sun, V. Fremont (2023). Object Tracking with a Fusion of Event-Based Camera and Frame-Based Camera. In: Arai, K. (eds) Intelligent Systems and Applications. IntelliSys 2022. Lecture Notes in Networks and Systems, vol 543. Springer, Cham. https://doi.org/10.1007/978-3-031-16078-3_15

ii. H. Sun, M. -Q. Dao and V. Fremont, "3D-FlowNet: Event-based optical flow estimation with 3D representation," 2022 IEEE Intelligent Vehicles Symposium (IV), Aachen, Germany, 2022, pp. 1845-1850, doi: 10.1109/IV51971.2022.9827380.

iii. S. Chen, H. Sun and V. Frémont, "Mono-Vision based Moving Object Detection using Semantic-Guided RANSAC," 2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Karlsruhe, Germany, 2021, pp. 1-6, doi: 10.1109/MFI52462.2021.9591165.

iv. S. Chen, H. Sun and V. Frémont, "A Semantic-Guided LiDAR-Vision Fusion Approach for Moving Objects Segmentation and State Estimation," 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China, 2022, pp. 4308-4313, doi: 10.1109/ITSC55140.2022.9922443.

Resumé

Selon un rapport de la National Highway Traffic Safety Administration (NHTSA), 94 % des accidents de la route sont dus à des erreurs humaines. Face à ce problème, des systèmes de conduite autonome sont en cours de développement pour prévenir les accidents et accroître l'efficacité du trafic.

Les percées dans le domaine de la vision par ordinateur sont le début de l'apprentissage profond et la disponibilité de nouveaux capteurs, tels que le lidar. En outre, l'augmentation de l'intérêt du public et du potentiel du marché a entraîné le développement de systèmes de conduite autonome avec différents degrés d'automatisation. Cependant, la conduite automatisée robuste en milieu urbain n'a pas encore été réalisée. Les accidents causés par des systèmes immatures sapent la confiance.

La conduite entièrement autonome dans des scénarios urbains est considérée comme le plus grand défi dans ce domaine depuis les premières tentatives. Lors du DARPA Urban Challenge en 2007, de nombreux groupes de recherche internationaux ont testé leurs systèmes de conduite autonome dans un environnement d'essai calqué sur une scène urbaine typique. Seules six équipes ont réussi à terminer l'épreuve, bien que cette compétition ait été l'événement le plus important et le plus significatif jusqu'alors. L'environnement de test était dépourvu de certains participants d'une scène de conduite urbaine réelle, comme les piétons et les cyclistes. Cependant, le fait que six équipes aient réussi à relever le défi a attiré l'attention.

Les systèmes de conduite autonome les plus modernes utilisent un large éventail de capteurs embarqués. Une redondance élevée des capteurs est importante dans la plupart des tâches pour la robustesse et la fiabilité. Le matériel peut être classé en cinq catégories : les capteurs extéroceptifs pour la perception, les capteurs proprioceptifs pour le contrôle de l'état interne, les unités de calcul, les réseaux de communication et les actionneurs. Les capteurs extéroceptifs sont principalement utilisés pour percevoir l'environnement, qui comprend des objets dynamiques et statiques, par exemple des zones carrossables, des obstacles et des piétons. Les caméras, les lidars, les radars et les capteurs à ultrasons sont les modalités les plus couramment utilisées pour cette tâche.

Les caméras peuvent détecter la couleur et sont passives puisqu'elles n'émettent au-

cun signal pour les mesures. La détection de la couleur est tout à fait nécessaire pour des tâches telles que la reconnaissance des feux de circulation. Plus important encore, la vision par ordinateur en 2D est un domaine bien établi avec de nombreux algorithmes de pointe. Cependant, les caméras présentent certains inconvénients. Les conditions d'éclairage affectent considérablement leurs performances, la vitesse de transmission fixe entraîne des problèmes de flou de mouvement, et les informations de profondeur sont difficiles à obtenir à partir d'une seule caméra. Des études prometteuses ont été menées pour améliorer la perception de la profondeur à l'aide d'une caméra monoculaire, mais des modalités qui ne sont pas affectées par l'éclairage et les mouvements rapides restent nécessaires pour les tâches de conduite dynamique. D'autres types de caméras suscitent un intérêt croissant, notamment les caméras à flash, les caméras thermiques et les caméras événementielles.

Les caméras événementielles font partie des modalités de détection les plus récentes utilisées dans les systèmes de conduite autonome. Contrairement aux caméras classique, qui acquièrent des images à une fréquence fixe (par exemple, 30 images par seconde), les caméras à événement répondent aux changements de luminosité de chaque pixel de la scène de manière asynchrone et indépendante. Ainsi, la sortie d'une caméra événementielle est une séquence dynamique d'"événements" numériques, chaque événement représentant un changement de luminosité d'un seuil prédéfini pour un pixel. Ce codage s'inspire de la nature de la stimulation des voies visuelles biologiques.

Les caméras événementielles sont des capteurs de données : leur sortie dépend de la variation de luminosité de la scène ou de la quantité de mouvement (mouvement propre et objets en mouvement). Plus le mouvement est rapide, plus le nombre d'événements générés est important, car chaque pixel adapte son taux d'échantillonnage au taux de changement. Les événements sont horodatés avec une résolution de l'ordre de la microseconde et peuvent être transmis avec une latence inférieure à la milliseconde, ce qui permet aux caméras événementielles de réagir rapidement aux stimuli visuels.

Les caméras événementielles présentent de nombreux avantages potentiels par rapport aux caméras classique :

1. Haute résolution temporelle : la surveillance des changements de luminosité est rapide dans les circuits analogiques, et la lecture des événements est numérique, généralement avec une horloge de 1 MHz, de sorte que les événements sont détectés et horodatés avec une résolution de l'ordre de la microseconde. Par conséquent, les caméras événementielles peuvent capturer des mouvements rapides sans souffrir de problèmes de flou de mouvement.

2. Faible latence : chaque pixel fonctionne indépendamment, et il n'est pas nécessaire d'attendre un temps d'exposition global. Dès que le changement est détecté, il est transmis. Les caméras événementielles ont donc une latence minimale : environ dix µs sur la paillasse du laboratoire et moins d'une milliseconde dans le monde réel.

3. Faible consommation d'énergie : Comme les caméras événementielles ne transmettent que les changements de luminosité et éliminent donc les données redondantes, l'énergie n'est utilisée que pour traiter les pixels changeants. Les systèmes de caméras embarquées où le capteur est directement interfacé avec un processeur ont montré une consommation d'énergie au niveau du système (c'est-à-dire détection plus traitement) de 100 mW ou moins.

4. Gamme dynamique élevée : La plage dynamique élevée des caméras événementielles (140 dB) dépasse largement les 60 dB des caméras à base d'images, ce qui leur permet d'acquérir des informations du clair de lune à la lumière du jour. Cela est dû au fait que les photorécepteurs des pixels fonctionnent sur une échelle logarithmique, et que chaque pixel travaille indépendamment, sans attendre un obturateur global.

Pour conclure les points mentionnés ci-dessus, pour concevoir un système de conduite autonome sûr et robuste, il faut sélectionner plusieurs capteurs pour le système de perception, et les caméras classique sont l'un des capteurs les plus populaires, malgré leurs inconvénients. Les caméras événementielles sont les capteurs les plus récents qui peuvent compenser les inconvénients des caméras classique pour le système de conduite autonome. Il est donc important d'étudier la vision basée sur les événements et la fusion entre les caméras classique et à événement.

Une question critique concernant les caméras événementielles est de savoir comment extraire des informations significatives des données événementielles pour remplir la tâche de perception. Il s'agit d'une question globale puisque la réponse dépend de l'application et qu'elle détermine la conception algorithmique du résolveur de tâches. Les caméras événementielles acquièrent des informations de manière asynchrone et éparse, avec une haute résolution temporelle et une faible latence. Par conséquent, l'aspect temporel joue un rôle important dans le traitement des événements. Selon le nombre d'événements traités simultanément, deux classes d'algorithmes peuvent être déterminées :

1. les méthodes qui fonctionnent sur la base d'un événement par événement, où l'algorithme traite l'événement un par un, ce qui rend les données de l'événement asynchrones.

2. les méthodes qui fonctionnent sur des groupes d'événements, qui introduisent une latence. Sans tenir compte des problèmes de latence, les méthodes basées sur des groupes d'événements peuvent toujours fournir un processus événement par événement si la fenêtre glisse d'un événement. Par conséquent, la différence entre les deux classes est plus subtile : un événement seul ne fournit pas suffisamment d'informations pour l'estimation, et des informations supplémentaires, sous la forme d'événements passés ou de connaissances supplémentaires, sont donc nécessaires.

En fonction de la manière dont les événements sont traités, on peut distinguer les approches basées sur des modèles et celles basées sur l'apprentissage profond. Chaque catégorie présente des méthodes avec des avantages et des inconvénients, et la recherche actuelle se concentre sur l'exploration des possibilités de chaque méthode.

De nombreux algorithmes concernant la vision basée sur les événements ont été développés au cours des dernières années. Selon la tâche abordée, on trouve la détection et le suivi d'éléments, l'estimation du flux optique, la reconstruction 3D (monoculaire et stéréo), l'estimation de la pose et le SLAM, l'odométrie visuo-inertielle, la reconstruction d'images, la segmentation du mouvement, la reconnaissance et le contrôle neuromorphique.

Pour l'état de la recherche et tous les problèmes mentionnés ci-dessus, nous attirons l'attention sur l'estimation du flux optique, la détection et le suivi d'objets avec les caméras événementielles, et la fusion entre les caméras image et événementielles. Les principaux travaux réalisés dans cette thèse sont résumés comme suit :

1. L'analyse du mouvement est l'un des problèmes les plus fondamentaux et les plus difficiles de la vision par ordinateur. L'estimation du flux optique est l'un des sujets qui représentent le mouvement des pixels d'images adjacentes. La caméra événementielle peut fournir des informations temporelles riches car les données événementielles contiennent des informations temporelles plus précises que l'image. En outre, grâce à sa plage dynamique élevée et à sa faible latence, la caméra événementielle peut mieux fonctionner dans des scénarios extrêmes. Pour extraire les informations de mouvement des événements, nous avons analysé les méthodes de prédiction du flux optique à partir des événements et proposé notre approche, qui atteint des performances de pointe.
2. Le suivi d'objets est une tâche essentielle dans le domaine de la vision par ordinateur. Dans la vue de la caméra, le problème du suivi peut être défini comme l'estimation de la trajectoire des objets dans le plan de l'image lorsqu'ils se dépla-

cent dans une scène. Pour obtenir un suivi d'objet meilleur et plus robuste, nous avons proposé un approche basée sur un modèle pour la détection et le suivi d'objet basé sur un modèle en utilisant un hybride de la caméra basée sur les événements et de la caméra basée sur les images. Nous avons d'abord proposé un algorithme de détection d'objets basé sur un modèle pour la caméra basée sur les événements. Nous avons ensuite développé une stratégie de fusion pour les caméras basées sur les événements et sur les images. Enfin, un filtre PMBM est adopté pour le suivi en utilisant la détection fusionnée à partir de l'hybride de la caméra basée sur l'événement et de la caméra classique.

3. Pour un véhicule autonome, il est nécessaire d'estimer complètement le modèle de mouvement de chacun des participants environnants et de planifier les trajectoires de l'ego en fonction de leurs états futurs afin de réduire les risques de collision. En raison du mouvement du véhicule ego et des contraintes liées à la formation de l'image, il est très difficile de classer les objets environnants comme étant en mouvement ou statiques, car même les objets statiques seront perçus comme étant en mouvement. La segmentation du mouvement implique que les deux tâches doivent être effectuées conjointement. La première se concentre sur la segmentation des objets, dans laquelle les objets de classes spécifiques intéressantes sont mis en évidence, comme les piétons ou les véhicules. La seconde se concentre sur la classification du mouvement, dans laquelle un classificateur prédit si l'objet observé est en mouvement ou statique. Comme mentionné précédemment, les caméras traditionnelles basées sur les images ne peuvent pas fournir d'informations temporelles, et les caméras basées sur les événements manquent d'informations sur l'apparence et la cohérence spatiale. Ces deux caractéristiques sont toutes deux importantes pour la segmentation des mouvements. Nous avons proposé un classique d'apprentissage profond pour la fusion entre la caméra basée sur l'événement et la caméra classique et réaliser la segmentation du mouvement.

En résumé, la thèse a proposé trois algorithmes de vision basés sur des événements qui tentent de fournir une meilleure et plus robuste perception des systèmes de conduite autonome. Les tests et analyses réalisés ont prouvé la faisabilité et la fiabilité des algorithmes de perception proposés.

Abstract

According to a report by the National Highway Traffic Safety Administration (NHTSA), 94% of road accidents are caused by human errors. Against this issue, autonomous driving systems are being developed to prevent accidents and increase traffic efficiency.

The breakthroughs in computer vision are the beginning of deep learning and the availability of new sensors, such as lidar. Furthermore, an increase in the public interest and market potential caused the development of autonomous driving systems with varying degrees of automation. However, robust automated driving in urban environments has yet to be achieved. Accidents caused by immature systems undermine trust.

Fully autonomous driving in urban scenarios has been seen as the biggest challenge in the field since the earliest attempts. During the DARPA Urban Challenge in 2007, many global research groups tried their autonomous driving systems in a test environment modeled after a typical urban scene. Only six teams managed to complete the event, although this competition was the biggest and most significant event up to that time. The test environment lacked certain participants of a real-world urban driving scene, such as pedestrians and cyclists. However, the fact that six teams managed to complete the challenge attracted significant attention.

State-of-the-art autonomous driving system utilize a wide selection of onboard sensors. High sensor redundancy is important in most tasks for robustness and reliability. Hardware can be categorized into five: exteroceptive sensors for perception, proprioceptive sensors for internal state monitoring, computational units, communication arrays, and actuators. Exteroceptive sensors are mainly used for perceiving the environment, which includes dynamic and static objects, e.g., drivable areas, obstacles, and pedestrians. Camera, lidar, radar, and ultrasonic sensors are this task's most commonly used modalities.

The cameras can sense color and are passive since they do not emit any signal for measurements. Sensing color is quite necessary for tasks such as traffic light recognition. More critically, 2D computer vision is an established field with many state-of-the-art algorithms. However, cameras have certain drawbacks. Illumination conditions affect their performance drastically, fixed transmit rate causes motion blur problems, and depth information is challenging to obtain from a single camera. There are promising studies to

improve monocular camera-based depth perception, but modalities that are not negatively affected by illumination and fast motion are still necessary for dynamic driving tasks. Other camera types gaining interest include flash cameras, thermal cameras, and event cameras.

Event-based cameras are among the newer sensing modalities that have been used in autonomous driving systems. In contrast to frame cameras, which acquire frame images at a fixed rate (e.g., 30 fps), event cameras, respond to brightness changes for each pixel in the scene asynchronously and independently. Thus, the output of an event camera is a dynamic rate sequence of digital "events", with each event representing a change of brightness of a predefined threshold at a pixel. This encoding is inspired by the spiking nature of biological visual pathways.

Event-based cameras are data-driven sensors: their output relies on the scene's brightness change or amount of motion (self-motion and moving objects). The faster the movement, the more events are generated since each pixel adapts its sampling rate to the rate of change. Events are timestamped with microsecond resolution and can be transmitted with sub-millisecond latency, which makes event cameras react quickly to visual stimuli.

Event cameras present multiple potential advantages over frame cameras:

1. High temporal Resolution: monitoring brightness changes is fast in analog circuitry, and the read-out of the events is digital, generally with a 1 MHz clock, so events are detected and timestamped with microsecond resolution. Therefore, event cameras can capture fast motions without suffering from motion blur problems.

2. Low latency: each pixel works independently, and there is no need to wait for a global exposure time. When the change is detected, it is transmitted. Hence, event cameras have minimal latency: about ten µs on the lab bench and sub-millisecond in the real world.

3. Low power consumption: Because event cameras transmit only brightness changes and thus remove redundant data, power is only used to process changing pixels. Embedded event-camera systems where the sensor is directly interfaced with a processor have shown system-level power consumption (i.e., sensing plus processing) of 100 mW or less.

4. High Dynamic Range: The high dynamic range of event cameras (140 dB) notably surpasses the 60 dB of frame-based cameras, allowing them to acquire information from moonlight to daylight. This is because photoreceptors of the pixels operate

on a logarithmic scale, and each pixel works independently, not waiting for a global shutter.

To conclude the points mentioned above, to design a safe and robust autonomous driving system, one should select multiple sensors for the perception system, and frame cameras are one of the most popular sensors, although the drawbacks. Event cameras are the newer sensors that can make up the drawbacks of the frame cameras for the autonomous driving system. So, it is important to investigate event-based vision and fusion between frame and event cameras.

One critical question of event cameras is how to extract meaningful information from the event data to fulfill the perception task. This is a comprehensive question since the answer is application dependent, and it drives the algorithmic design of the task solver. Event cameras acquire information asynchronously and sparsely, with high temporal resolution and low latency. Hence, the temporal aspect plays an important role in processing events. Depending on how many events are processed simultaneously, two classes of algorithms can be determined:

1. methods that work on an event-by-event basis, where the algorithm process the event one by one, thus keeping asynchronous of the event data.
2. methods that work on groups of events, which introduce latency. Discounting latency concerns, methods based on groups of events can still provide an event-by-event process if the window slides by one event. Therefore, the difference between both classes is more subtle: an event alone does not provide enough information for estimation, and so additional information, in the form of past events or extra knowledge, is needed.

Depending on how events are processed, we can distinguish between model-based approaches and deep learning-based approaches. Each category presents methods with advantages and disadvantages, and current research focuses on exploring each method's possibilities.

Many algorithms concerning event-based vision have been developed in the past few years. According to the task addressed, there are feature detection and tracking, optical flow estimation, 3D reconstruction (monocular and stereo), pose estimation and SLAM, visual-inertial odometry, image reconstruction, motion segmentation, recognition, and neuromorphic control.

For the research status and all the problems mentioned above, we draw attention to optical flow estimation, object detection and tracking with the event cameras, and fusion

between frame and event cameras. The main work carried out in the thesis is summarized as follows:

1. Motion analysis is one of the most fundamental and challenging problems in computer vision, the optical flow estimation is one of the topics that represent the pixel motion of adjacent frame images. The event-based camera can provide rich temporal information because the event data contains more accurate temporal information than the frame image. And because of the high dynamic range and low latency, the event camera can work better in extreme scenarios. To extract the motion information from events, we analyzed the methods for optical flow prediction from events and proposed our approach, which achieves state-of-the-art performance.
2. Object tracking is an essential task within the field of computer vision. In the camera view, the tracking problem can be defined as estimating the trajectory of objects in the image plane as it moves around a scene. To achieve a better and more robust object tracking, we proposed the framework for model-based object detection and tracking using the hybrid of the event-based and frame-based camera. We first proposed a model-based object detection algorithm for the event based camera. Then a fusion strategy for the event-based and frame-based cameras was developed. At last, a PMBM filter is adopted for tracking using the fused detection from the hybrid of the event-based and frame-based camera.
3. For an autonomous vehicle, it is required to fully estimate the motion model of each of the surrounding participants and to plan the ego-trajectories based on their future states to reduce collision risks. Due to the ego vehicle's motion and constraints related to image formation, it is very challenging to classify the surrounding objects as moving or static because even static objects will be perceived as moving. Motion segmentation implies that the two tasks have to be performed jointly. The first focuses on object segmentation, in which objects of specific interesting classes are highlighted, such as pedestrians or vehicles. The second focuses on motion classification, in which a classifier predicts whether the observed object is moving or static. As mentioned previously, traditional frame-based cameras cannot provide temporal information, and event-based cameras lack appearance information and spatial consistency. The two features are both important for Motion segmentation. We proposed a deep learning framework for the fusion between the event-based and frame-based camera and realize Motion segmentation.

To sum up, the thesis proposed three event-based vision algorithms that try to provide a better and more robust perception of autonomous driving systems. The tests and analyses conducted proved the feasibility and reliability of the proposed perception algorithms.

Table of Contents

List of Figures

List of Tables

GLOSSARY

AV Autonomous Vehicle. 77

CED Color Event Camera Dataset. 28

CNN Convolutional Neural Network. 31, 36, 41

CT Contrast Threshold. 26

DARPA Defense Advanced Research Projects Agency. 6, 11

DDD20 DAVIS Driving Dataset 2020. 28

DS Direction Selective. 37

DSEC A Stereo Event Camera Dataset for Driving Scenarios. 28, 47

EM Expectation Maximisation. 31

ESS Event-based Semantic Segmentation. 30

HOTA Higher Order Tracking Accuracy. 74

HS Horn and Schunck. 33–35

IMU Inertial measurement unit. 41

LIF Leaky Integrate-and-Fire. 45, 46

LK Lucas and Kanade. 34, 35

MOT Multi-Object Tracking. 60

MVSEC Multi Vehicle Stereo Event Camera Dataset. 28, 48, 53

NHTSA National Highway Traffic Safety Administration. 6, 11

PMBM Poisson multi-Bernoulli mixture. 10, 14, 18, 29, 32, 60, 61, 65, 74

RFS Random Finite Sets. 60, 65

SAE Surface of Active Events. 37

SLAM Simultaneous localization and mapping. 9, 13, 27, 29

SNN Spiking Neural Network. 44–46

SNR Signal-to-noise ratio. 60

STDP Spike-timing-dependent plasticity. 46

UAVs unmanned aerial vehicles. 25

UDA unsupervised domain adaptation. 30

Chapter 1

Introduction

1.1 Overview and context of the work

Monocular frame-based cameras are one of the most commonly used sensors for object-tracking tasks in many different areas, such as autonomous vehicles [84], unmanned aerial vehicles (UAVs) [23], and industrial robots [49]. It synchronously transmits images, frame by frame at a fixed rate. It has main drawbacks such as low temporal resolution, redundant information and low dynamic range. More importantly, a frame image is transmitted at a fixed frequency and does not contain time information, which is important for the object tracking issue. More recently, the event-based camera, a bio-inspired technology of silicon retinas, has been proposed to solve both classical and new computer vision tasks [7], [33]. These cameras are asynchronous sensors that monitor the brightness change of each pixel related to the viewed scene with a precise timestamp, which means that the event-based camera provides explicitly time information. Thus, event-based cameras have a large potential for computer vision applications in challenging scenarios compared to standard cameras. Typically, they are used on the sensing modalities such as unmanned aerial vehicles (UAVs) [60], robotics [11] or wearable electronics [12], where operation is under unrealistic lighting conditions and sensitive to the temporal resolution. The common applications for event-based vision are object tracking [11], surveillance and monitoring [35], and optical flow estimation [1], [92]. For autonomous driving, [41] proposed a method that can predict the vehicle's steering angle according to the event data, and [5], [25] proposed datasets that contain event data along with the vehicle control and diagnostic data.

The event-based camera was first designed in the early 1990s with pioneering work on Silicon Retina and Address Event Representation (AER) by Misha Mahowald and Carver Mead. The motivation was to build the brain in silicon [17] and mimic biological vision's spiking, asynchronous nature. Compared to conventional frame-based cameras, biological retinas have many advantages, such as high efficiency, low power consumption,

Figure 1.1 – The frame-based camera fails in extreme scenarios

and low latency, and these advantages could become available to artificial retinas. The event-based cameras rely on the smart pixels to generate events that can track their measured brightness and trigger events when reaching the threshold and reset, adjusting dynamically to changes in luminance. So event-based cameras are also called Dynamic Vision Sensors.

The first practical and robust event-based camera is the Dynamic Vision Sensor (DVS) [33] developed in 2008. This work caused the product of the DAVIS-240 and later the DAVIS-346 cameras, with a 240×180 and 346×240 spatial sensor resolution, respectively. A valuable feature of the two sensors was that they provide not only the event data but also frame images on the same chip. This feature allowed recording both events and conventional frames, perfectly registered and synchronized with each other. This feature, in combination with well-documented development software packages, resulted in the widespread adoption of the DAVIS cameras in the research community. As a result, they are the most commonly cited event-based camera hardware in the literature despite competition from large industry efforts led by manufacturers such as Prophesee. Recently, Prophesee has improved its development support by providing easy-to-use Application Programming Interfaces (APIs). And they released an event camera with a resolution of 1280×720 [16], but still without synchronized frame images.

For an event-based camera, each pixel keeps detecting the change in the measured intensity. When the change in log intensity from the previous measurement exceeds the predefined Contrast Threshold (CT), an event is triggered and transmitted immediately. Each event is in the format of a tuple, consists of the event location $[x, y]$, polarity $p \in \{-1, +1\}$ (indicating the direction of the brightness change), and timestamp t (which is the precise timestamp of the event in μs resolution). Because each pixel can report events at any time, independently of the other pixels, the event based camera can take

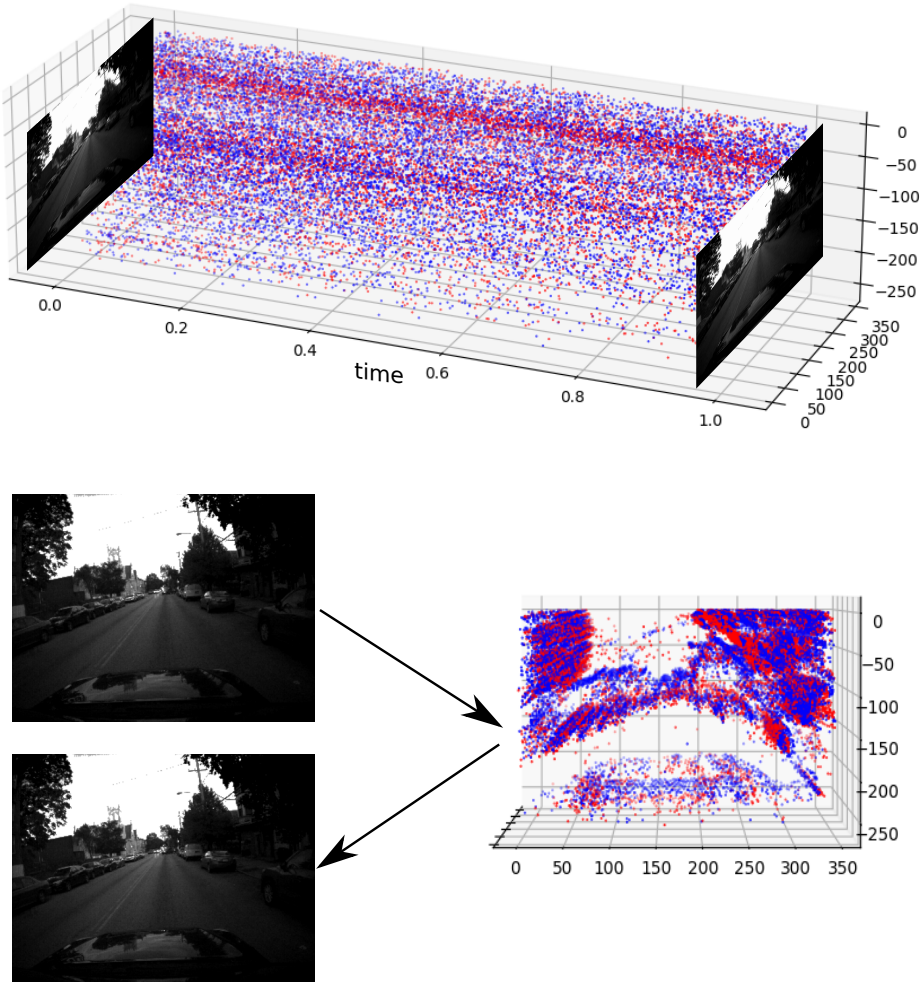


Figure 1.2 – The visualization of the event data

samples at a dynamic and scene-dependent rate. Since event-based cameras only give updates on brightness changes in the scene, they are an efficient means of encoding visual information. The frame-based cameras under-sample the scene when changes occur faster than the fixed frame-rate, resulting in motion blur problems. The event-based cameras avoid this by sampling the scene at a dynamic optimal rate regarding per-pixel brightness changes.

Since event data is spatial-temporal data extended by polarity, it is better to view it in a three-dimensional spatio-temporal plot, as shown in the figure (1.2). Each event is represented by its location (x, y), timestamp t, and polarity p. From this point of view, event data is much more similar to the point cloud than frame images.

Datasets	Resolution	Purpose
IJRR [45]	240×180	Pose Estimation, Visual Odometry, and SLAM
MVSEC [91]	346×260	Depth and Optical Flow
CED [61]	346×260	Image Reconstruction
DDD20 [25]	346×260	Steering Prediction
DSEC [20]	640×480	Semantic, Depth and Optical Flow

Table 1.1 – Event-based camera datasets and their purpose

There are two main ways to process events: model-based and model-free (also known as data-driven or deep-learning-based) approaches. Due to the significant success that deep-learning-based approaches have achieved for traditional computer vision using frame cameras and LiDAR, there is now growing research interest in applying deep learning techniques to event cameras, such as 3D reconstruction [9], [90], Pose Estimation and SLAM [73], Image Reconstruction [52], [62], and Recognition [19], [87]

For bench-marking and training the event-based neural network models, several datasets are proposed for the event-based vision task, such as SLAM, depth and optical flow estimation, and image reconstruction. The IJRR [45] event-based camera dataset was recorded with the DAVIS240C model. It has a comparatively low spatial resolution of 240×180 pixels and lengthens just about 20 minutes. However it has 13 gigabytes (event data and frames images), and only 19 % is used up by the concurrent frame images because of the fast-moving scenes. DDD20 [25], which are more compact than the frame-based representation, contain comparatively slow-moving scenes. Table (1.1) shows the information for various event camera datasets.

Modern vehicles are equipped with various sensors to perceive the surrounding environment; each sensor has its advantages and disadvantages. For example, radar provides good performance of depth measurement for close obstacles, but they lack semantic information and perform poorly for far objects. Frame-based cameras, instead, provide rich appearance information from which scene semantics can be extracted; however, they lack temporal information and rely on scene illumination. Their performances are highly degraded in adverse illumination conditions and fast-motion scenarios. On the other hand, the event cameras provide accurate temporal information and do not suffer from bad illumination and fast motion but lack appearance information. Data fusion has been proven to provide improved performance in various tasks [27], [54], [64], [70]. So, it is important to fuse the frame-based and event-based camera information to achieve a more robust perception system for autonomous vehicles.

1.2 Aims, Contributions and outline of thesis

Event-based cameras are naturally suitable for tasks that describe motion or moving objects, such as optical flow estimation, tracking, or motion segmentation. Such descriptions of motion are essential parts of the autonomous driving system, as they are related to tasks such as detection, SLAM, visual odometry, or depth estimation. This thesis aims to hybrid the event-based camera and frame-based camera to achieve more robust object detection and tracking for the autonomous driving. Specifically, we aim to:

1. Extract the motion information using the event data
2. Develop object detection algorithms for event cameras.
3. Develop fusion strategies between the event-based camera and frame-based camera.
4. Develop object tracking algorithms using a hybrid of the event camera and the frame camera.

we list our contributions and the outline of this thesis. Key contributions are listed, following a brief introduction of the chapter following.

Chapter 2

Chapter 2 analyzes the methods for optical flow prediction from events. The key contributions of this chapter are:

- Analyze and compare the current scene flow algorithms for event based cameras.
- A deep learning method is proposed for event based cameras to predict scene flow.

To kickstart our process, we prioritize optical flow estimation using an event camera. Optical flow describes the apparent motion pattern of objects in an image over time, resulting from either object movement or self-motion. Precise optical flow estimation is essential for tasks that involve motion or moving objects. The event camera outperforms frame images by providing more accurate optical flow, as event data contains more precise temporal information. Moreover, the event camera's high dynamic range and low latency enable more accurate optical flow computation even in extreme scenarios. First, we introduce the model-based optical flow algorithm utilized in our object tracking framework. Subsequently, we present a deep learning model [70] capable of predicting accurate optical flow, achieving state-of-the-art performance.

Chapter 3

Using the model-based optical flow method discussed in Chapter 2, we can proceed to the object detection and tracking stage. In Chapter 3, we proposed the framework

for model-based object detection and tracking using the hybrid of the event-based and frame-based camera. The key contributions of this chapter are:

- The model-based object detection algorithm for the event based camera.
- The fusion strategy between the event-based and frame-based camera.
- The adapted Poisson multi-Bernoulli mixture (PMBM) filter for tracking with a hybrid of the event-based and frame-based camera.

This chapter presents a framework [71] that leverages both event-based and frame-based cameras to enhance object tracking effectiveness. The framework employs a clustering algorithm to generate detections for the event data and a fusion strategy to combine the frame camera detection. For tracking, we utilize a Poisson multi-Bernoulli mixture (PMBM) filter. By utilizing the benefits of both camera types, our framework achieves more robust object tracking.

Chapter 4

As the model-based framework is complex and computationally inefficient, in this chapter, we apply deep learning to the problems of moving object detection and fusion with the frame based camera. The main contributions are:

- A deep learning framework to detect moving objects.
- A deep learning framework for the fusion between the event-based and frame-based camera.

The primary contribution of our work is an end-to-end deep learning model that facilitates moving object detection and fusion between event and frame cameras. Our model takes an end-to-end approach, enabling simultaneous moving object detection and sensor fusion. Our work harnesses the benefits of fusing events and images to conduct motion segmentation in low illumination environments. By leveraging this fusion approach, we significantly improve the detection of moving objects and achieve state-of-the-art performance.

Chapter 2

Camera-based Optical flow analysis

2.1 Introduction

Motion analysis is one of the most fundamental and challenging problems in computer vision, the optical flow estimation is one of the topics that represent the pixel motion of adjacent frame images. Before discuss the optical flow estimation with the event cameras, we need to introduce the optical flow estimation with the frame cameras first. Because most of the event-based algorithms are developed based on the frame cameras' approach. The first framework for optical flow estimation is proposed by Horn and Schunck (HS) [24], which uses a variational method to estimate dense optical flow (2.4). The objective can be seen as a global energy function that is then minimized. Let the image be $p = (x, y)^T$, and the underlying flow field is $w(p) = [u(p), v(p)]^T$, where u and v are the optical flow field. The HS algorithm needs two assumptions:

1. smoothness in the flow over the whole image
2. brightness constancy assumption

With first the assumption, we can have the brightness constancy equation:

$$\begin{aligned} I(x,y,t) =& I(x+dx, y+dy, t+dt) \\ & \Downarrow \\ \frac{\partial I}{\partial x}\frac{dx}{dt} + & \frac{\partial I}{\partial y}\frac{dy}{dt} = -\frac{\partial I}{\partial t} \end{aligned} \tag{2.1}$$

Where $(\frac{dx}{dt}, \frac{dy}{dt})^T$ is the optical flow $(u, v)^T$, $(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})^T$ and $\frac{\partial I}{\partial t}$ is the spatial and temporal gradient $(I_x, I_y)^T$ and I_t, so the brightness constancy equation can also be written as:

$$I_x u + I_y v + I_t = 0 \tag{2.2}$$

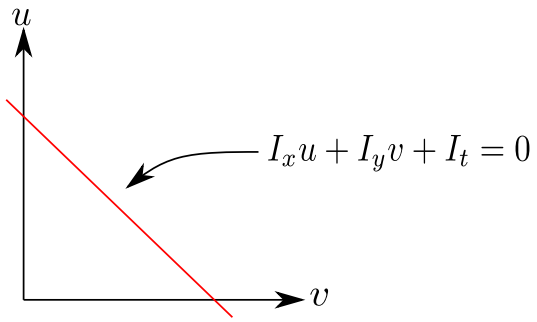


Figure 2.1 – Solution with the brightness constancy constraint

With the Eq. (2.2), we can achieve the first part of the objective function:

$$\min_{u,v}[I_x u_{i,j} + I_y v_{i,j} + I_t]^2 \tag{2.3}$$

However, with a single constraint, the solution cannot be determined as shown in the figure (2.1), the solution lies on a straight line. So, we need to consider the second assumption, smoothness. This assumption requires the optical flow is smooth from pixel to pixel and the constraint is shown in figure (2.2). Combine the two constraints can the final constraint function can be written as:

$$\min_{u,v} \sum_{i,j}[E_s(i,j) + \lambda E_d(i,j)]^2 \tag{2.4}$$

Where E_d is the brightness constancy constraint, and E_s is the smoothness constraint:

$$\begin{aligned} E_d(i,j) &= [I_x u_{i,j} + I_y v_{i,j} + I_t]^2 \\ E_s(i,j) &= \frac{1}{4}[(u_{ij} - u_{i+1,j})^2 + (u_{ij} - u_{i,j+1})^2 + (v_{ij} - v_{i+1,j})^2 + (v_{ij} - v_{i,j+1})^2] \end{aligned} \tag{2.5}$$

After the HS method, Lucas and Kanade (LK) [37] introduce local constraints and estimate the sparse flow field following the assumptions:

1. small displacement within the neighbour field

2. approximately constant within a neighborhood of the point under consideration

If we consider a 5 x 5 image patch, 25 equations can be obtained according to the brightness

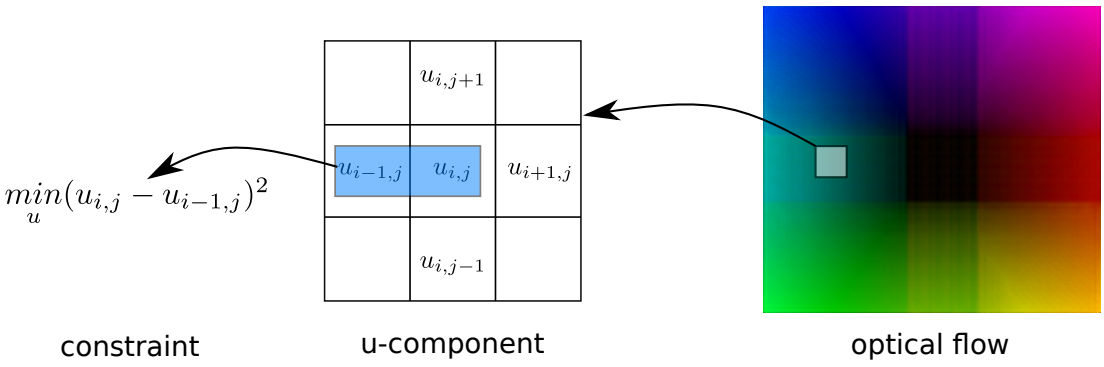


Figure 2.2 – Smooth flow constraint for the u-component of the optical flow

constancy equation in Eq. (2.1):

$$\begin{aligned} I_x(p_1)u + I_y(p_1)v &= -I_t(p_1) \\ I_x(p_2)u + I_y(p_2)v &= -I_t(p_2) \\ &\vdots \\ I_x(p_{25})u + I_y(p_{25})v &= -I_t(p_{25}) \end{aligned} \tag{2.6}$$

where $p_1, p_2, ..., p_{25}$ is the 25 points in the image patch. We can also write the Eq. (2.6) into the matrix form:

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_{25}) \end{bmatrix} \tag{2.7}$$

Then the least squares solution solve is:

$$\begin{bmatrix} \sum_{p\in P} I_x I_x & \sum_{p\in P} I_x I_y \\ \sum_{p\in P} I_y I_x & \sum_{p\in P} I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum_{p\in P} I_x I_t \\ \sum_{p\in P} I_y I_t \end{bmatrix} \tag{2.8}$$

Based on the basic methods HS and LK, many improvements and modifications have been proposed [15], [50], [67], [81]; to solve several significant problems in the model-based methods: large displacement, occlusion, and illumination change.

In terms of motion analysis, the event-based camera has the potential to achieve better

performance for this task than the frame-based camera. Because the former captures motion information while the latter records static absolute pixel value. Also, due to the high dynamic range and temporal resolution, the event-based camera is free of motion blur problems and can work in extreme light conditions. Therefore, developing the optical flow algorithms for the event-based camera is essential. However, event based optical flow estimation is not without challenges. The "brightness constancy" assumption central to the classical models of Horn-Schunck and Lucas-Kanade does not hold in event data space. The deep learning models are also challenging because the event data is asynchronous and in a format of a tuple.

To overcome the challenges, in this chapter, we will discuss both the model-based and deep learning-based approaches for optical flow estimation using an event-based camera. We will first introduce the model-based optical flow algorithm that we adopted in our model-based object detection and tracking framework. After that, we will propose our deep learning approach [70], which achieves better performance compared to the state-of-the-art approach.

2.2 Model-based approach

2.2.1 Related work

In the event-based vision, optical flow algorithms for event cameras have three levels because of the asynchronous data type. First is the feature level, where optic flow is estimated only at select parts of the image plane. Second is event masked dense, which means the optical flow is estimated for each event; for the pixel that does not have event data, there will have no optical flow estimations. The third is fully dense, where optical flow is estimated at each pixel.

An early optical flow algorithm was presented by [10], in which the bio-inspired Direction Selective (DS) filters were used. The Direction Selective filter is based on the Surface of Active Events (SAE); the SAE means an image in which each pixel contains the timestamp of the most recent event at that location. The Direction Selective filter compares the timestamp of the incoming event to the previous events in the 0°, 45°, and 90° orientation. Because the edges of moving object fire neighboring events almost instantaneously, a small difference in timestamp between neighbors implies an edge, which can be compared to prior edge detections in the SAE to compute the velocity through time-of-flight. Inspired

by the [10], [48] proposed a spiking neural network model to simulate several banks of DS filters. [8] use Direction Selective Gabor filter banks to estimate optic flow and apply normalization to ameliorate the aperture problem.

However, this method only allows for discretized directions; [3] improved this work by adapting the famous Lukas-Kanade algorithm. They maintained a histogram of previous events over a short time window and updated it with subsequent events. The histogram is used to estimate spatial and temporal gradients, which serve as input to an overdetermined set of equations based on brightness constancy, which can then be solved for the optic flow vector by least squares.

[59] further improved this work by providing symmetric gradients using central finite differences. Since the spatio-temporal derivatives of event images are used, this method is relatively sensitive to noise. This is because the event images may carry on small values due to the potentially small number of events triggered by edges and the natural sparsity of event data. This causes the derivative estimates to be unreliable. To reduce this effect, [59] employed a Savitsky-Golay filter to smooth the event image.

The following works [4] proposed methods that fit a plane to each event on the SAE and calculate the local gradient. This gradient is then equivalent to the optical flow of the event at that location. Since this planar fit only approximates the surface at a given point, the assumption of constant velocity in the local neighborhood is required. The planar fit is calculated with the equation: $ax + by + ct + d = 0$ and requires at least three events (the event plus the local neighbors).

2.2.2 Adopted Approach

In this section, we will introduce the work of [1] because it can provide accurate dense optical flow estimations and is naturally synchronized with frame images. This method will be used in our model-based object detection and tracking framework.

[1] proposed a novel optical flow method that leverages the high spatial fidelity of synchronous image frames and the high temporal resolution of event data. They compute the spatial derivative $(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})$ from frame images as the usual way and calculate the $\frac{\partial I}{\partial t}$ from the event data. Next, we will introduce the details of this approach.

First, let $J(\mathbf{x}, t)$ denote a *log* intensity image sequence:

$$J(\mathbf{x}, t) = log(I(\mathbf{x}, t)) \tag{2.9}$$

Then we can define event data as when the change in log intensity exceeds a predefined threshold τ :

$$\begin{aligned} t_{i+1}(\mathbf{x}) &= \arg\min_{t}\{t > t_i \Big| |J(\mathbf{x}, t_i) - J(\mathbf{x}, t)| > \tau\} \\ p_{i+1}(\mathbf{x}) &= sign(J(\mathbf{x}, t_{i+1}) - J(\mathbf{x}, t_i)) \end{aligned} \tag{2.10}$$

Here, the event is in the format of tuple $(\mathbf{x}, t_{i+1}, p_{i+1})$, where $\mathbf{x}$ is the position of the pixel, t_i is the precise timestamp of when the event occurs, p_i is the polarity indicating whether the change in the pixel intensity was brighter or darker.

Then, we need to adapt the brightness constancy equation (2.1) to log the brightness constancy equation by dividing both sides by $I(\mathbf{x}, t)$:

$$\frac{\partial J(\mathbf{x}, t)}{\partial x}\frac{dx}{dt} + \frac{\partial J(\mathbf{x}, t)}{\partial y}\frac{dy}{dt} = -\frac{\partial J(\mathbf{x}, t)}{\partial t} \tag{2.11}$$

Given the relations:

$$(\frac{\partial J(\mathbf{x}, t)}{\partial x}, \frac{\partial J(\mathbf{x}, t)}{\partial y})^T = \frac{(\frac{\partial I(\mathbf{x}, t)}{\partial x}, \frac{\partial I(\mathbf{x}, t)}{\partial y})^T}{I(\mathbf{x}, t)} \tag{2.12}$$

$$\frac{\partial J(\mathbf{x}, t)}{\partial t} = \frac{\frac{\partial I(\mathbf{x}, t)}{\partial t}}{I(\mathbf{x}, t)} \tag{2.13}$$

If we compute the spatial derivative $(\frac{\partial I(\mathbf{x}, t)}{\partial x}, \frac{\partial I(\mathbf{x}, t)}{\partial y})^T$ from gray images $I(\mathbf{x}, t)$ with the usual way and compute $\frac{\partial I(\mathbf{x}, t)}{\partial t}$ from $\frac{\partial J(\mathbf{x}, t)}{\partial t}$, then we can write the brightness constancy equation as follows:

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} = -\frac{\partial J(\mathbf{x}, t)}{\partial t} \cdot I(\mathbf{x}, t) \tag{2.14}$$

With the equation (2.14), we now are able to calculate the $\frac{\partial J(\mathbf{x}, t)}{\partial t}$ with the event data. Assuming for the moment that log pixel intensity value $J(\mathbf{x}, t)$ is known at times $\{t_i(\mathbf{x})), t_{i+1}(\mathbf{x})), ..., t_N(\mathbf{x})\}$. Assume further that we can recover $J(X, t)$ for an arbitrary $t \in R$ using the N_{th} degree polynomial interpolation, with the equation:

$$J(\mathbf{x}, t) = \sum_{n=0}^{N} c_n t^n \tag{2.15}$$

The coefficients $\mathbf{C} = (c_0, c_1, ..., c_N)$ can be obtained from solving a system of equations

as follows:

$$\underbrace{\begin{pmatrix} J(\mathbf{x}, t_i(\mathbf{x})) \\ J(\mathbf{x}, t_{i+1}(\mathbf{x})) \\ \vdots \\ J(\mathbf{x}, t_{i+N}(\mathbf{x})) \end{pmatrix}}_{\mathbf{J}} = \underbrace{\begin{pmatrix} 1 & t_i(\mathbf{x})^1 & \dots & t_i(\mathbf{x})^N \\ 1 & t_{i+1}(\mathbf{x})^1 & \dots & t_{i+1}(\mathbf{x})^N \\ \vdots & & & \vdots \\ 1 & t_{i+N}(\mathbf{x})^1 & \dots & t_{i+N}(\mathbf{x})^N \end{pmatrix}}_{\mathbf{A}} \cdot \underbrace{\begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_N \end{pmatrix}}_{\mathbf{C}} \tag{2.16}$$

Taking derivative of (2.15) we can get the estimation of the $\frac{\partial J(\mathbf{x},t)}{\partial t}$:

$$\frac{\partial J(\mathbf{x}, t)}{\partial t} = \sum_{n=1}^{N} n c_n t^{n-1} \tag{2.17}$$

With equation (2.17), we can calculate the $\frac{\partial J(\mathbf{x},t)}{\partial t}$ with the coefficients $\{c1, ..., c_N\}$. Recalling that we have the assumption that $J(\mathbf{x}, t)$ is known at times $\{t_i(\mathbf{x})), t_{i+1}(\mathbf{x})), ..., t_N(\mathbf{x})\}$, which is not in practice. Therefore we cannot solve the equation (2.17) directly, and we need to calculate the coefficients $\{c1, ..., c_N\}$ differently:

With the definition of the event data, we can accumulate the events to recursively reconstruct $J(\mathbf{x}, t_{i+N})$ from $J(\mathbf{x}, t_i)$, as follows::

$$J(\mathbf{x}, t_{i+N}) = J(\mathbf{x}, t_i) + \tau \sum_{n=1}^{N} p_{i+n} \tag{2.18}$$

Here, τ is the predefined threshold.

Then, we can rewrite the $\mathbf{J}$ as:

$$\begin{aligned} &\underbrace{\begin{pmatrix} J(\mathbf{x}, t_i(\mathbf{x})) \\ J(\mathbf{x}, t_{i+1}(\mathbf{x})) \\ \vdots \\ J(\mathbf{x}, t_{i+N}(\mathbf{x})) \end{pmatrix}}_{\mathbf{J}} = \begin{pmatrix} J(\mathbf{x}, t_i(\mathbf{x})) \\ J(\mathbf{x}, t_i(\mathbf{x})) + \tau p_{i+1}(\mathbf{x}) \\ \vdots \\ J(\mathbf{x}, t_i(\mathbf{x})) + \tau(p_{i+1}(\mathbf{x}) + \cdots + p_{i+N}(\mathbf{x})) \end{pmatrix} \\ =& J(\mathbf{x}, t_i(\mathbf{x})) \underbrace{\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}}_{\mathbf{1}} + \tau \underbrace{\begin{pmatrix} 0 \\ J(\mathbf{x}, t_i(\mathbf{x})) + p_{i+1}(\mathbf{x}) \\ \vdots \\ J(\mathbf{x}, t_i(\mathbf{x}) + \tau(p_{i+1}(\mathbf{x}) + \cdots + p_{i+N}(\mathbf{x}) \end{pmatrix}}_{\mathbf{P}} \end{aligned} \tag{2.19}$$

With equations (2.16) and (2.19), the coefficients $\mathbf{C} = (c_0, c_1, ..., c_N)$ can be computed as:

$$\mathbf{C} = \mathbf{A}^{-1}\mathbf{J} = J(\mathbf{x}, t_i(\mathbf{x}))\mathbf{A}^{-1}\mathbf{1} + \tau\mathbf{A}^{-1}\mathbf{P} \tag{2.20}$$

In Eq. (2.20), $\mathbf{A}^{-1}\mathbf{1} = (1, 0, ..., 0)^T$, because the first column of $\mathbf{A}$ in Eq. (2.16) is $\mathbf{1}$ as shown in Eq. (2.19). Therefore, the log pixel intensity $J(\mathbf{x}, t_i)$ has no influence over $\{c1, ..., c_N\}$. After all, we can calculate the coefficients $\{c1, ..., c_N\}$ as:

$$\begin{pmatrix} c_1 \\ \vdots \\ c_N \end{pmatrix} = \tau\mathbf{A}^{-1+} \begin{pmatrix} 0 \\ p_{i+1}(\mathbf{x}) \\ \vdots \\ p_{i+1}(\mathbf{x}) + \cdots + p_{i+N}(\mathbf{x}) \end{pmatrix} \tag{2.21}$$

With the calculated $\frac{\partial J(\mathbf{x},t)}{\partial t}$, we can then obtain the optical flow with the equation (2.14). This algorithm leverages the high temporal resolution of the event data to compute the temporal gradient at each frame and exploits the spatial fidelity of frame images to estimate the spatial gradient, resulting in reliable estimates of motion vectors. We will use it to provide optical flow estimations in our model based object detection and tracking framework.

Figure (2.1) shows the optical flow estimation of [1] on various scenarios. The yellow arrow is the optical flow estimation, and the red arrow is the reference optical flow calculated from the Inertial measurement unit (IMU) data according to Eq. (2.22), ignoring the translation. All the flow vectors are downsampled and rescaled for visualization. The estimations concentrated on the edges and contours of objects where the brightness changes frequently. Because only the brightness changes, we can obtain enough event data to estimate the optical flow.

$$\begin{aligned} \frac{dx}{dt} &= -\omega_y + \omega_z Y + \omega_x XY - \omega_y X^2 \\ \frac{dy}{dt} &= +\omega_x - \omega_z X - \omega_y XY + \omega_x Y^2 \end{aligned} \tag{2.22}$$

where the $(\omega_x, \omega_y, \omega_z)$ is vector of the rotational velocities in radians/second.

In Figure (2.1a,b,c), we can confirm the accuracy of the optical flow by comparing them with the IMU flow. These three sequences are recorded with only angular motion so that the IMU flow can provide a trustable reference to the calculated results. We do not

calculate the IMU flow in Figure (2.1e,g,i) because there are moving objects in the scene, and there are many translation motions in the scenes. In the next chapter, the calculated optical flow will be used in our object detection methods for the event cameras.

Just as Convolutional Neural Network (CNN) reached a new performance for the frame-based camera, recent works have done the same in event-based vision. So in the next section, we will propose our own method based on the deep learning approach.

2.3 Deep-learning-based approach

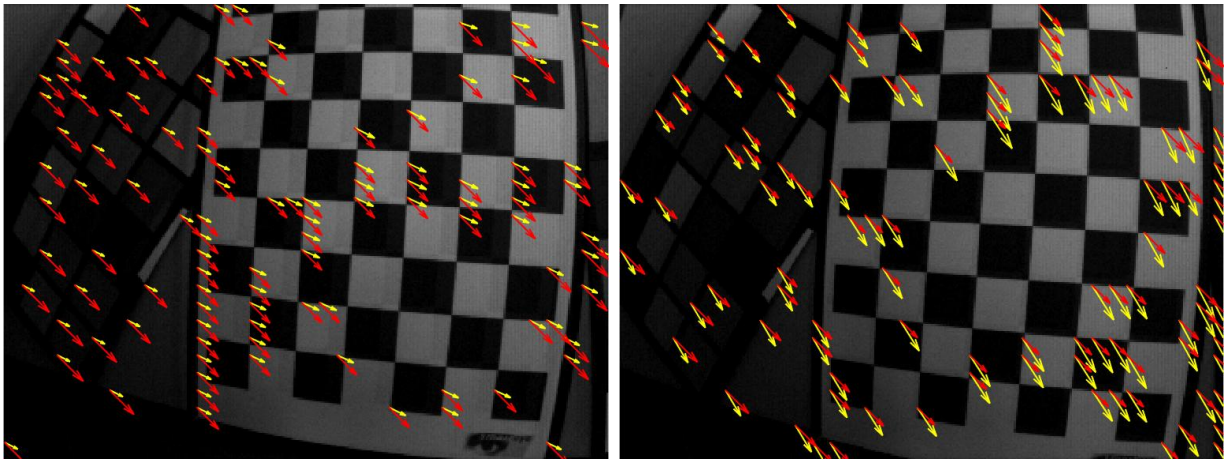
Although many works have already focused on using deep neural networks to estimate the optical flow for frame images, they cannot directly be adapted to the event-based vision. There are two main challenges for developing a deep learning-based approach for the event-based camera. Firstly, the event data is asynchronous and sequential, while the deep neural networks take the matrix-like data. An extra data preprocessing step is needed so it can be fed into the neural network. Also, because the event-based camera is new compared to the traditional sensors, the datasets, especially those with the ground truth, are insufficient. However, neural network training depends on the quantity and quality of the dataset. So another challenge is to train the network with less ground truth data.

2.3.1 Related work

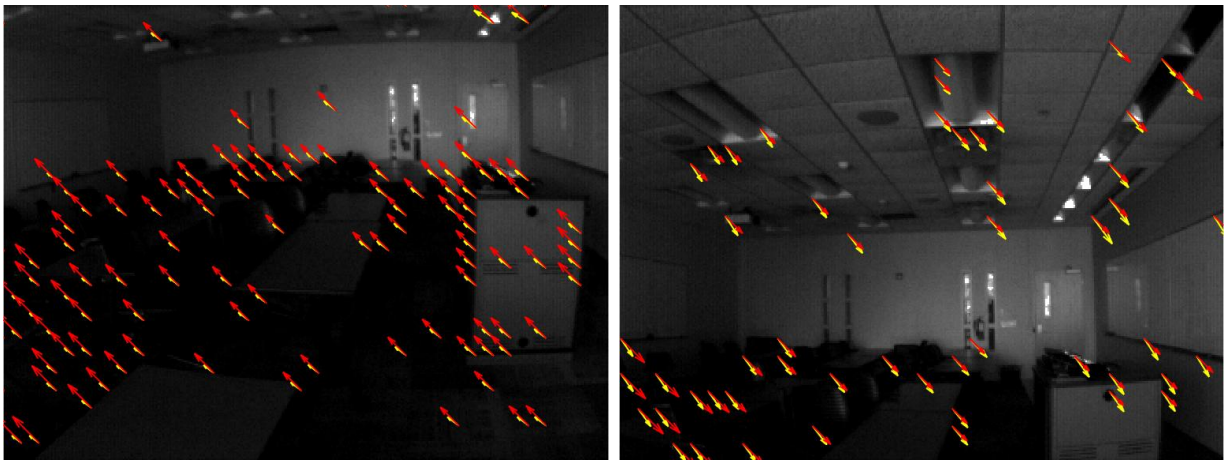
For the optical flow with the frame-based camera, Convolutional Neural Network (CNN) have been successfully used to solve the optical flow problem. Existing architectures for data-driven methods can be divided into two types: U-Net-based and spatial pyramid networks.

The first work is proposed by [14] called FlowNet, which has an encoder-decoder structure similar to the U-Net. Based on this basic encoder-decoder architecture, many modifications [82], [88], [89] are proposed. [26] successively stack the FlowNet to create a large network called FlowNet2.0. The accuracy of optical flow estimation is improved due to the stacked iteration of sub-FlowNet. However, the model size and computation consumption increased hugely.

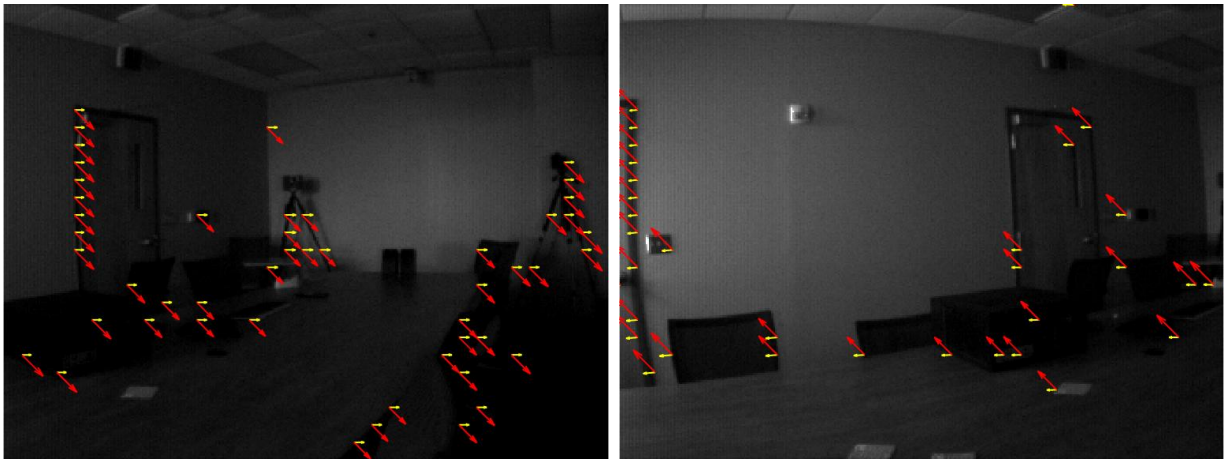
[53] first propose a coarse-to-fine spatial pyramid network named SPyNet that can output optical flow at multiple resolution levels. The main advantage of SPyNet is its small model size. However, their performance still cannot catch up with FlowNet2.0. The



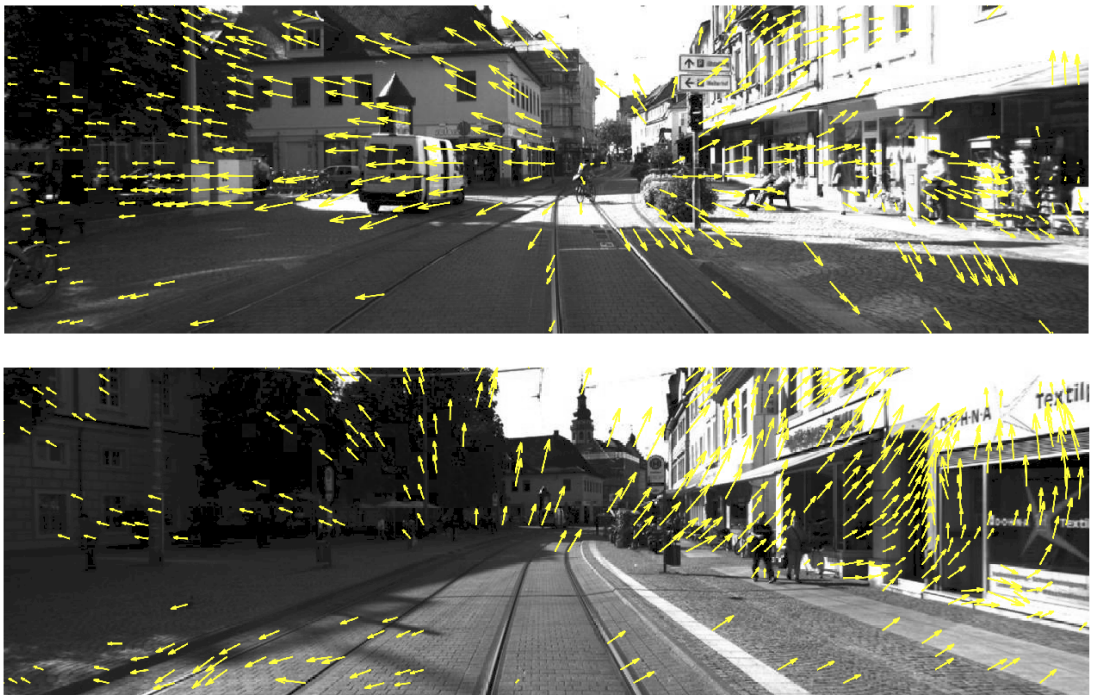
(a) checkerboard



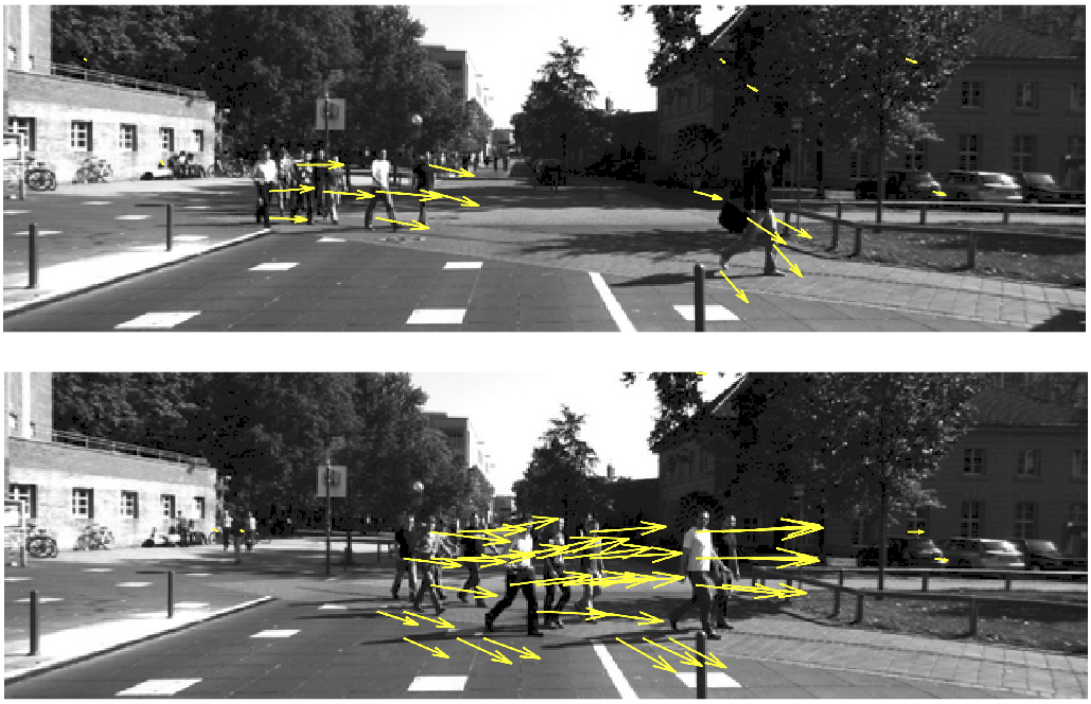
(b) classroom



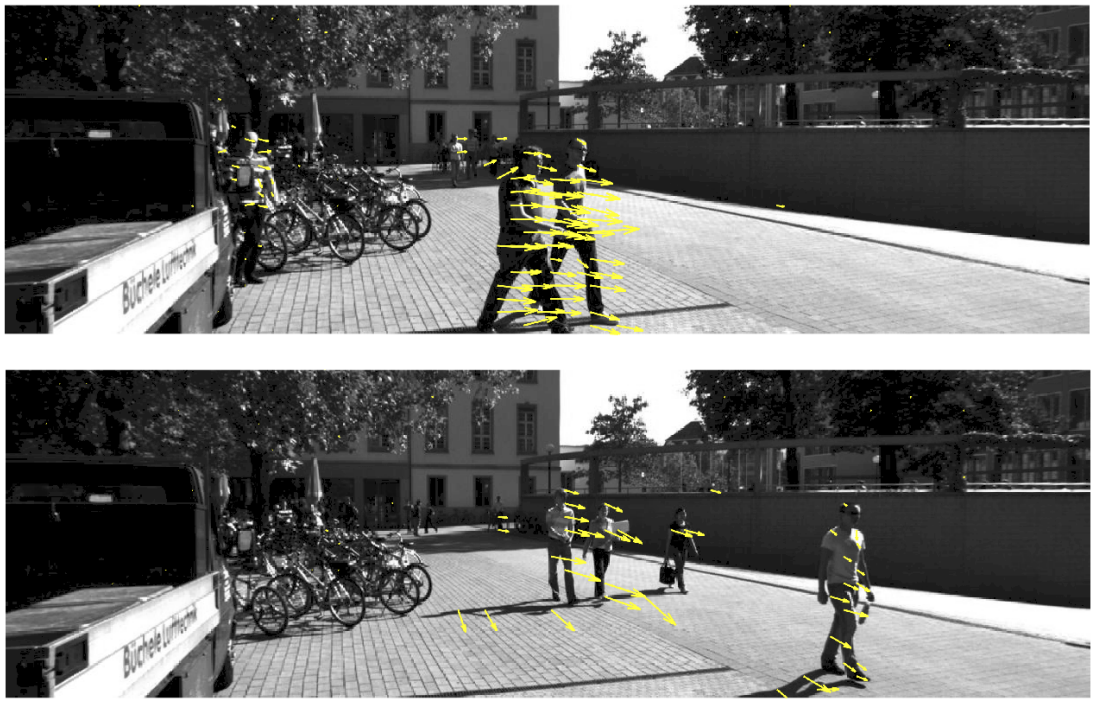
(c) conference room



(e) KITTI-13 sequence



(g) KITTI-16 sequence



(i) KITTI-17 sequence

Figure 2.1 – Optical flow estimation of [1] on various scenarios.

SPyNet estimates large motions on the coarse level and warps the second image toward the first using the up-sampled flow from the previous level. Thus, it only calculates residual flow at each level. Based on [53], [68] propose a modified spatial pyramid network named PWC-Net to deal with large displacements and decrease the number of parameters. They employ feature warping operation rather than image warping at different scale and uses the cost volume layer to calculate the matching cost at each pyramidal level, and achieves promising results on several public datasets.

For the event camera, the research can be divided into two categories regarding the challenges discussed above: Input adaption and Datasets for training.

Input Adaption

For the first problem, there are two categories to adapt the neural network algorithms to the event-based vision: Spiking Neural Network and encoding the event data to image-like representation for the traditional convolution neural network. Next, we will give an introduction to these two methods.

For the first category, the Spiking Neural Network (SNN), inspired by the biological

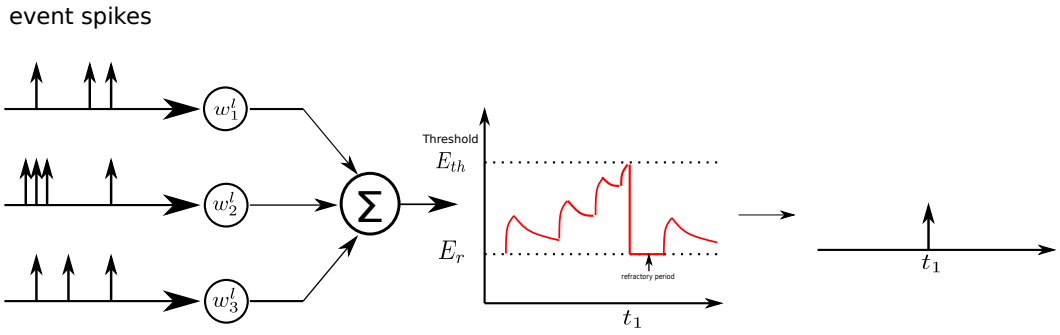


Figure 2.2 – The dynamics of the Leaky Integrate-and-Fire (LIF) neuron model. The input events are multiplied by the synaptic weight and then combined as the current inflow in the membrane potential. Whenever the membrane potential reaches the threshold, the neuron fires an output spike and resets the membrane potential to membrane rest potential.

neuron model, integrate the concept of time into their model. The neurons in the SNN do not transmit information at each propagation cycle but transmit information when a membrane potential – an intrinsic state of the neuron related to its current inflow – reaches the threshold. This mechanism allows them to represent and integrate different information dimensions such as time, frequency, and phase. So the SNN can naturally capture the spatio-temporal dynamics of the event data with the precise spike timings [72]. Due to the different inference mechanisms, the encoding of information through spikes is still an open issue. The Leaky Integrate-and-Fire (LIF) model is one of the popular spiking neuron models [28], which can be characterized by an internal state known as the membrane potential. The membrane potential accumulates the inputs over time and emits an output spike whenever it exceeds a set threshold. This mechanism naturally matches the event-based asynchronous data, which also has a precise timestamp. The LIF neuron model can be defined by the following equation:

$$\tau_m \frac{dv_m}{dt} = -(v_m(t) - E_r) + R_m I(t) \tag{2.23}$$

where $v_m(t)$ is the membrane potential, τ_m is the membrane time constant, E_r is the membrane rest potential which is a constant, R_m is the membrane resistance, and $I(t)$ is the sum of current supplied by the input synapses. The sum of current is then can be

calculated by:

$$I(t) = W \cdot S(t) \tag{2.24}$$

where $W = [w_1, w_2, ..., w_N]$ are the weights for the synapses connection. $S(t) = [s_1(t), s_2(t), ..., s_n(t)]$ is the input spike like the event data.

$$s_i(t) = \sum_{(x,y,p)} \delta(t - t_i^{(x,y,p)}) \tag{2.25}$$

where $t_i^{(x,y,p)}$ is the timestamp of the event at position (x, y) with polarity p.

The Leaky Integrate-and-Fire (LIF) model shows that the spiking neural network naturally fits the event data format. The spike generation function of an LIF neuron is a hard threshold function that emits a spike when the membrane potential exceeds a firing threshold. Due to this discontinuous and non-differentiable neuron model, standard backpropagation algorithms cannot be applied to SNN in their native form. Hence, several approximate methods [30], [32] have been proposed to estimate the surrogate gradient of the spike generation function. For the optical flow estimation using SNN, [51] presented a methodology for optical flow estimation using convolutional SNN based on Spike-timing-dependent plasticity (STDP) unsupervised training [13]. The problem of this work is that they employ shallow SNN architectures because deep SNN needs to improve performance. Also, the experiment results are only evaluated on relatively simple tasks. [32] have shown that the number of spikes vanishes at deeper layers, leading to performance degradations in deep SNN. [31] try to solve this problem by proposing a deep hybrid neural network architecture called Spike-FlowNet. However, the training of the Spiking Neural Network still needs to be faster and more stable.

For the second category, there are also several works that train a traditional neural network to process the event data by transforming the events into a grid-based representation. [92] summarize the number of events at each pixel and the last timestamp and average timestamp at each pixel into an image-like matrix. Then, they calculate the optical flow using an encoder-decoder network as shown in figure (2.3). However, this encoding method loses the information because of its way of summarization. [93] proposed the following work and tries to solve this problem by adopting a bilinear sampling kernel and using more channels for the event image so the temporal resolution increases, but their method relies on depth and ego-motion. [83] adopted the same encoding method. However, they estimate the optical flow via depth and ego-motion and assume the scene

is static.

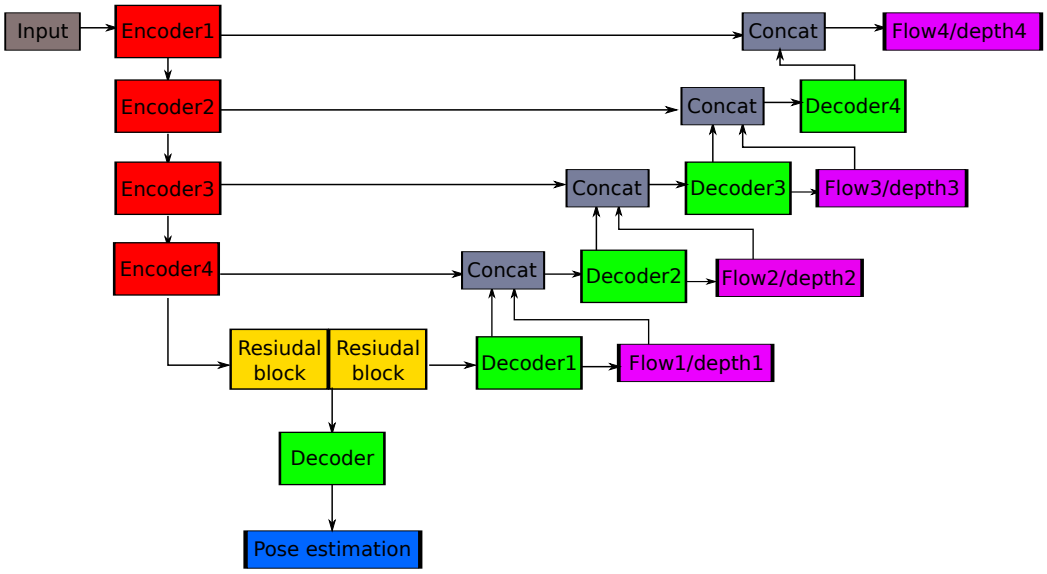


Figure 2.3 – Network structure of the [93].

Datasets and training

The second problem is the lack of datasets, especially those with ground truth. There are two ways to overcome this problem: record the dataset and generate the ground truth with the help of the Lidar and the frame camera or use a self-supervised training strategy.

For the event based optical flow dataset, [92] proposed the MVSEC dataset. They computed ground truth optical flow from Lidar odometry and mapping for the outdoor scenes. However, the ground truth quality is impaired by the lack of proper handling of occlusions and moving objects. To solve this problem, [20] proposed the DSEC dataset. They use stereo matching filtering to remove the Lidar points on moving objects. Furthermore, to ensure accurate optical flow ground truth, they manually inspect the whole dataset and extract all parts of sequences that contain static environments.

For the self-supervised training, [31], [92] use a self-supervised loss based on gray images to replace ground truth. [93] presents an unsupervised learning approach using only event camera data to estimate optical flow by accounting for motion blur in the event image and then learning to rectify the motion blur.

Contributions

The main contribution of this chapter is to propose a new encoding method and the corresponding neural network architecture to process an event data stream. We proposed a 3D encoding representation that can better preserve the temporal nature of the event data. We also present the 3D-FlowNet, a novel neural network architecture that can process the 3D input and generate optical flow estimations. Finally, we train and evaluate the proposed 3D-FlowNet using the Multi Vehicle Stereo Event Camera Dataset (MVSEC) [91]. The results show that our approach outperforms current state-of-the-art methods; we achieve 13% improvement compared to the Spike-FlowNet [31], and 32% compared to the EV-FlowNet [92].

2.3.2 Proposed Approach

In this section, we explain our approach in detail. We first explain our event encoding method, which encodes a group of event measurements into a 3D temporal-spatial event image. Then, we describe the architecture of our network, which uses 3D convolutions to process the spatial-temporal measurements and output the pixel-wise optical flow. Finally, we describe our training strategy, and the self-supervised loss is also discussed, this work has been published in [70].

Event Data Encoding Method

The event-based camera records the log intensity change of each pixel of the artificial retina and generates an event whenever the log intensity changes over the threshold θ:

$$log(I_{t+1}) - log(I_t) \geq \theta \tag{2.26}$$

The event measurement is in the format of a tuple, which consists of the location of the pixel, timestamp of the event, and polarity of the change:

$$e = (x, y, t, p) \tag{2.27}$$

Because the events are transmitted asynchronously, they cannot be immediately fed into standard convolutional neural network layers. It is therefore important to keep the necessary information while generating the encoding representation from the event stream.

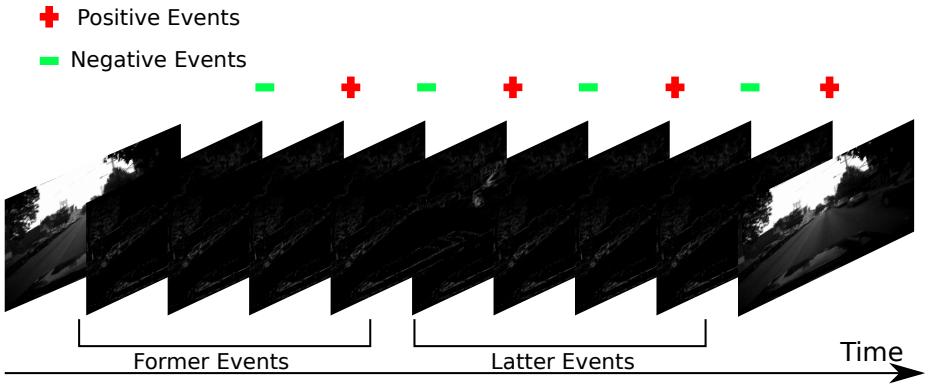
Several prior works have proposed different methods that transform the event output into a synchronous image-like representation. In EV-FlowNet [92], only the latest pixel-wise timestamps and the event counts are used to encode the event representation. However, fast motions and dense scenarios can enormously overlap per-pixel timestamp information. In [66], [93], the time domain is discretized to preserve the temporal distributions. To improve the resolution and the temporal domain beyond the number of bins, the authors insert events into this volume using a linearly weighted accumulation similar to bilinear interpolation. However, the number of input channels increases significantly as the time dimensions are finely discretized, further increasing the computation time for encoding and forward propagation.

Considering all the methods discussed before, we propose in this work, a novel input representation that can better exploit the information in the event data with less computation complexity. Given a set of N input events $E_N = (x_i, y_i, t_i, p_i), i \in [1, N]$, and a time depth D to discretize the time dimension of event data, we accumulate each group of event into images as follows:

$$\begin{aligned} t_{norm} &= (t - t_0)/(t_N - t_1) * (D - 1) \\ I(x, y, t, p) &= \sum_i \delta(p - p_i) k_b(x - x_i) k_b(y - y_i) k_b(t - t_{norm}) \\ k_b(a) &= max(0, 1 - |a|) \end{aligned} \tag{2.28}$$

Here, (x, y) denotes the position of the event, p is the polarity of the event, and δ is the Kronecker delta operator. $k_b(\cdot)$ denotes bi-linear sampling kernel. The generated event image I is a $(2, D, H, W)$ matrix, where the number 2 represents the positive and negative polarity, D is the discretized time depth, and (H, W) are respectively the height and width of the image. Then we split the event image into former and latter groups through the time dimension and obtained a new event image with the shape of $(4, \frac{D}{2}, H, W)$. Here the number 4 represents the four channels: Former positive events, former negative events, latter positive events, and latter negative events. Figure (2.4) shows the proposed input representation. Figure (2.4a) is the visualization of the event image and the relative grayscale image, left is one slice of the event image, and the brighter represents the more recent timestamp value. Figure (2.4b) is an example of the event representation where $D = 4$.

(a) Example of an event image and a gray scale image



(b) four channels for event data

Figure 2.4 – Visualization of our event encoding representation. (a) is one slice of the event image $I_{slice} = (1, 1, H, W)$, and the brighter represents the more recent timestamp value. (b) is an example of the event representation where $D = 4$.

Proposed Network Architecture

With the input representation $I_{4,D/2,H,W}$ discussed before, we propose the 3D-FlowNet architecture to predict the optical flow values. The 3D-FlowNet's network adopts an encoder-decoder architecture, containing four encoder layers, two residual blocks, and four decoder layers as shown in figure (2.5). First, the input event image is passed through two 3D-encoders. The 3D-encoders down-sample the time dimension $d/2$ to 1, and compress the 3D input into 2D ($(4, D/2, H, W) \rightarrow (4, 1, H, W) \rightarrow (4, H, W)$). Then the resulting activation are passed through two 2D-encoders, two residual blocks, and four 2D-decoders. For each decoder, the activation is up-sampled using the 2D transposed convolution and then convolved, to obtain the final optical flow estimation.

There is a skip connection from each encoder to the corresponding decoder. For the skip

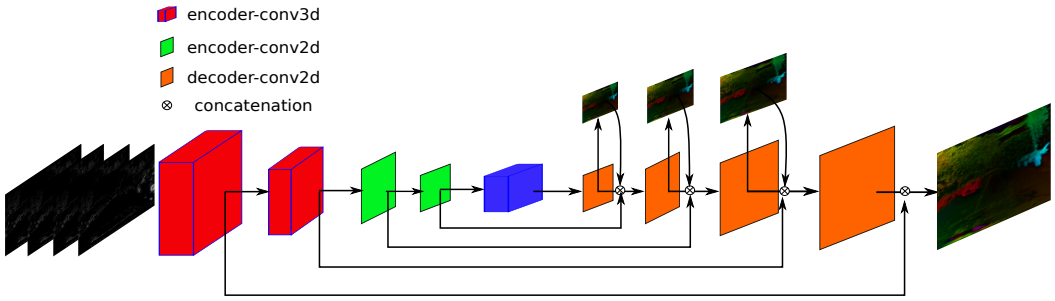


Figure 2.5 – Network structure of the 3D-FlowNet.

connection between 2D-encoder and 2D-decoder, the activation of the encoder is directly concatenated with the intermediate optical flow value and the activation of decoder. For the skip connection between 3D-encoder and 2D-decoder, the 3D activation ($C \times D \times W \times H$) is flattened into 2D tensor ($(C \cdot D) \times W \times H$) first, then it can be concatenated with the activation of the decoder and the intermediate optical flow. The predicted optical flows are then used together with the grayscale image for the loss calculation.

The details of the 3D-FlowNet are shown in figure (2.6). Red rectangle denotes the 3D convolutional modules, where [3DConv,a,b,c,(d_1, d_2, d_3), e] denotes a 2D convolutional layer with the kernel size a, stride b, output channel c, padding size (d_1, d_2, d_3) and normalization layer e (BN is Batch Normalization). The green rectangle denote the 2D convolutional block, where [2DConv,a,b,c,(d_1, d_2), e] denotes a 2D convolutional layer with the kernel size a, stride b, output channel c, padding size (d_1, d_2) and normalization layer e. Orange rectangle denote the transpose 2D convolutional block, where [2DConvT,a,b,c,(d_1, d_2), e] denotes a 2D transpose convolutional layer with the kernel size a, stride b, output channel c, padding size (d_1, d_2) and normalization layer e. The blue rectangle represent ResBlock.

Self-Supervised Loss

The event-based camera is a sensor that can produce synchronous grayscale images and asynchronous event data streams simultaneously. Compared to frame-based camera datasets, the number of available event-based camera datasets with annotated labels suitable for optical flow estimation is relatively small. As a result, for training our Spike-FlowNet, we used a self-supervised learning method that uses proxy labels from the recorded grayscale images [42], [85].

The total loss consists of a smoothness loss (L_{smooth}) and a photometric reconstruction

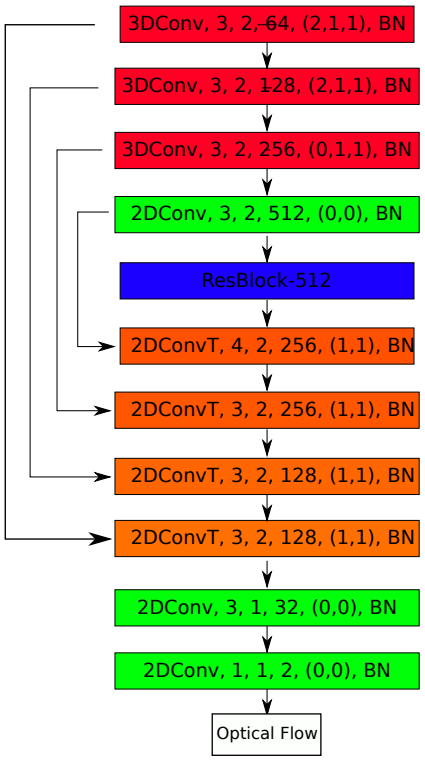


Figure 2.6 – The details of the 3D-FlowNet

loss (L_{photo}) [85]. The network needs a pair of grayscale images ($I_t, I_{t+\Delta t}$) to calculate the photometric loss, as well as the event data in the time window $(t, t + \Delta t)$. The second grayscale image is warped to the first grayscale image using the network's predicted optical flow. The photometric loss (L_{photo}) is used to minimize the difference between the first grayscale image and the inversely warped second grayscale image. This loss is based on the photometric consistency assumption, which states that a pixel value from the first image will be similar to the second frame warped by the predicted optical flow. The photometric loss can be written as:

$$\begin{aligned} &L_{loss}(u, v, I_t, I_{t+\Delta t}) = \\ &\sum_{x,y} \rho(I_t(x, y) - I_{t+dt}(x + u(x, y), y + v(x, y))) \end{aligned} \tag{2.29}$$

Then, the smoothness loss is adopted to improve the spatial consistency of neighboring optical flow. It is calculated as:

$$\begin{aligned} L_{smooth} = \sum_i \sum_j (&||u_{i,j} - u_{i+1,j}|| + ||(u_{i,j} - u_{i,j+1}||+ \\ &||(v_{i,j} - v_{i+1,j}|| + ||(v_{i,j} - v_{i,j+1}||) \end{aligned} \tag{2.30}$$

The total loss for the training is computed as the weighted sum of the photometric and smoothness loss:

$$L_{total} = L_{photo} + \lambda L_{smooth} \tag{2.31}$$

where λ is the weight factor.

2.3.3 Experiments

Dataset and Implementation Details

The MVSEC dataset [91] is used for training and evaluating optical flow predictions. The MVSEC dataset contains stereo event-based camera data, including flying, driving, and handheld scenes. Moreover, the dataset provides ground truth poses and depth maps for each event-based camera, and the ground truth optical flow can be generated accordingly. To offer fair comparisons with prior works [31], [92], only the outdoor day2 sequence is used for training.

During the training, the input is centrally cropped to 256×256 size. The ADAM optimizer is used, and the initial learning rate of 1e-4. The model is trained for 30 epochs

with a batch size of 16, while [31] takes 100 epochs. This is because the training of the ANN is faster and more stable than the SNN one.

Results

Here, the Average End-point Error (AEE) is used to evaluate the optical flow result, and it is defined as:

$$AEE = \frac{1}{n} \sum_{n} \|(u, v)_{pred} - (u, v)_{gt}\|^2 \tag{2.32}$$

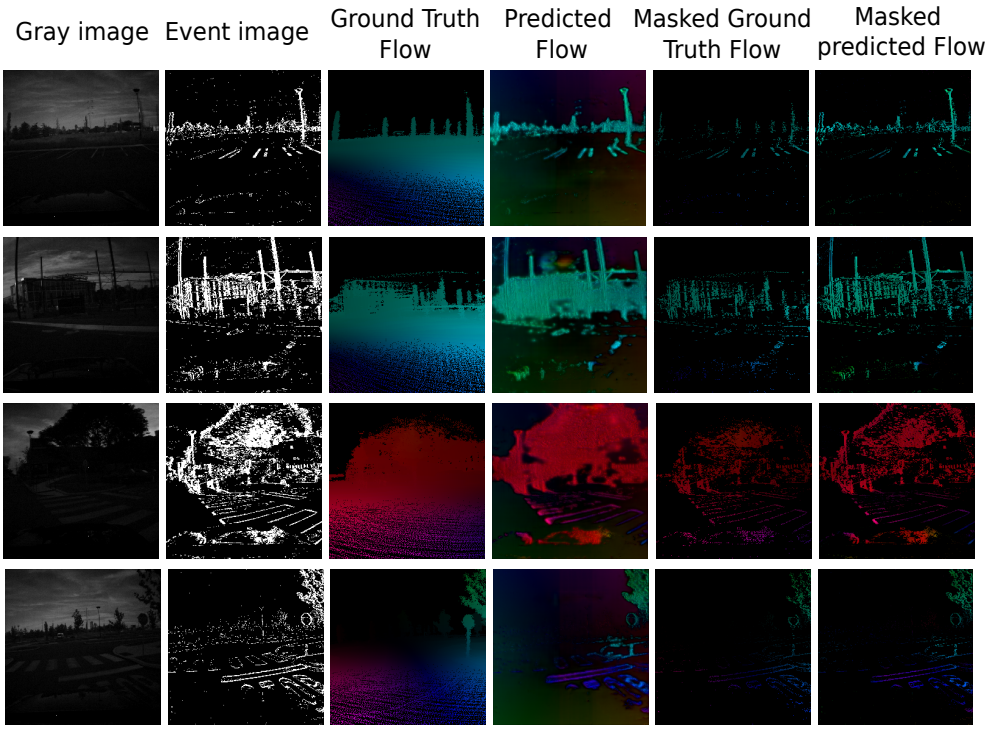
Table 2.1 – Quantitative assessment of our approach compared to EV-FlowNet and Spike-FlowNet

	outdoor day1		indoor flying1		indoor flying2		indoor flying3	
	AEE ↓	Outlier ↓	AEE ↓	Outlier ↓	AEE ↓	Outlier ↓	AEE ↓	Outlier ↓
EV-FlowNet [92]	0.49	0.2	1.03	2.2	1.72	15.1	1.53	11.9
Spike-FlowNet [31]	**0.49**	-	0.84	-	1.28	-	1.11	-
Ours	0.51	**0.1**	**0.7**	**0.1**	**1.10**	**0.2**	**0.91**	**0.1**

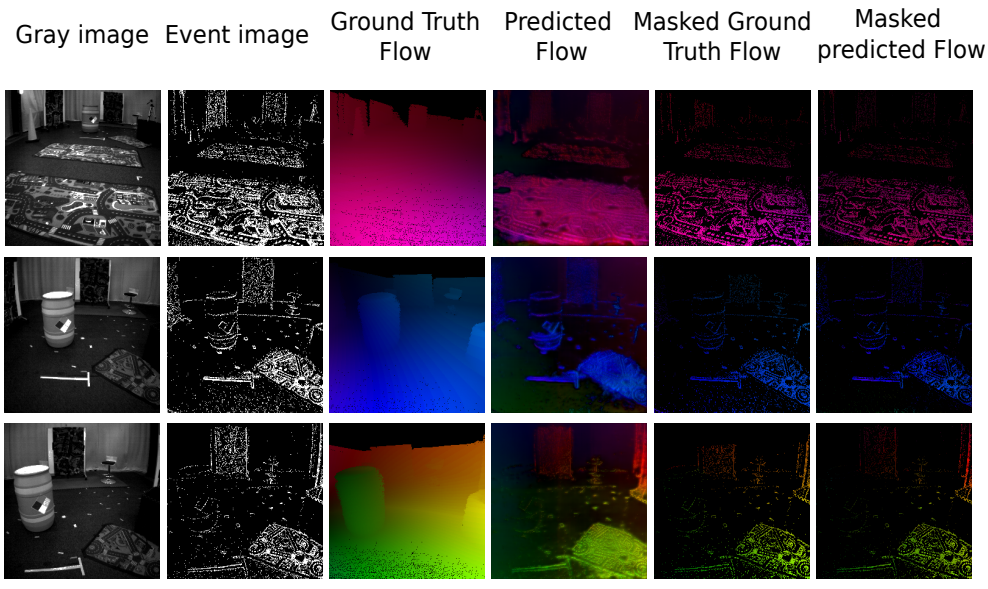
Where n is the number of the active pixel in the event image, $(u, v)_{pred}$ is the predicted optical flow and $(u, v)_{gt}$ is the groundtruth. We also count the outliers that corresponds to the percentage of points with AEE exceeding three pixels. For each sequence, the AEE is calculated in pixels, and the %Outlier is defined as the percentage of points with $AEE < 3$ pix. During the testing, the optical flow is also estimated on the centrally cropped 256×256 event images. The sequences of indoor flying 1,2,3 and outdoor day 1 are used. We use all events from the *indoor_flying* sequences and take events within 800 gray scale frames for the *outdoor_day1* sequence similar to [31].

Table (2.1) show the results of the AEE evaluation in comparison to previous event-based camera-based optical flow estimation approaches. Our approach achieves better performances than the others in all the *indoor_flying* sequences. Our AEE performance is similar to the others in the *outdoor_day1* sequence, but we obtain fewer outliers.

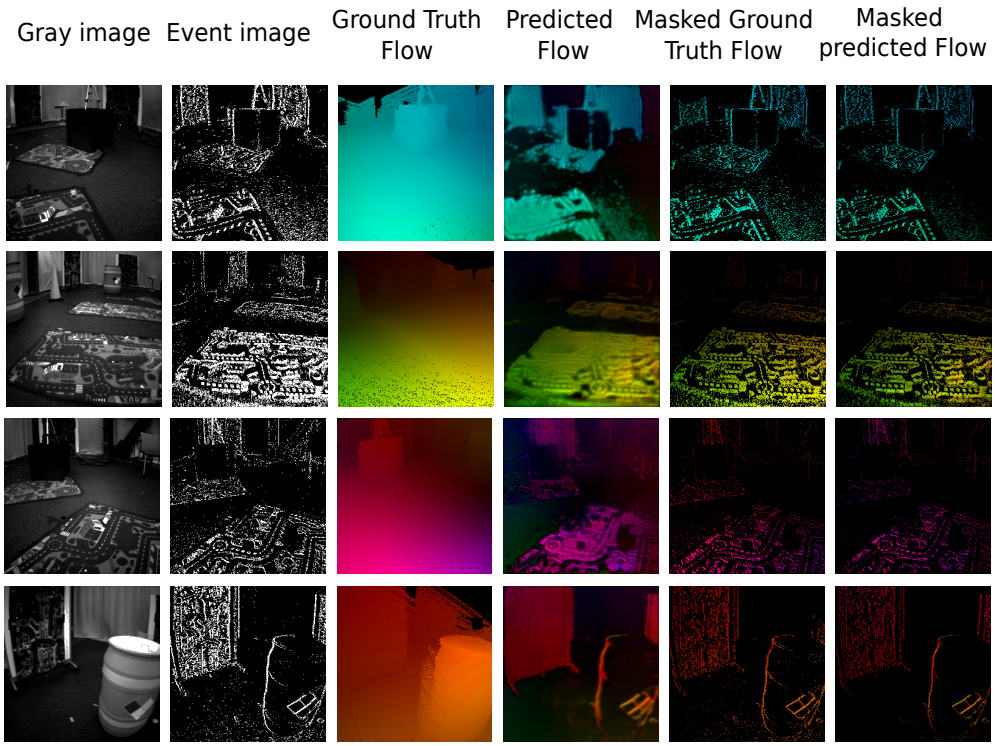
Figure (2.6) shows the qualitative results of our approach. The grayscale, event image, ground truth flow, and corresponding predicted flow images are displayed in this figure. Notice that the estimated optical flow is less dense than the ground truth. In the evaluation, we mask out the optical flow at points where the event data are absent. The masked optical flow is used here because event-based cameras detect the brightness change in pixels. Low texture regions, such as flat surfaces, produce very few events due to fewer



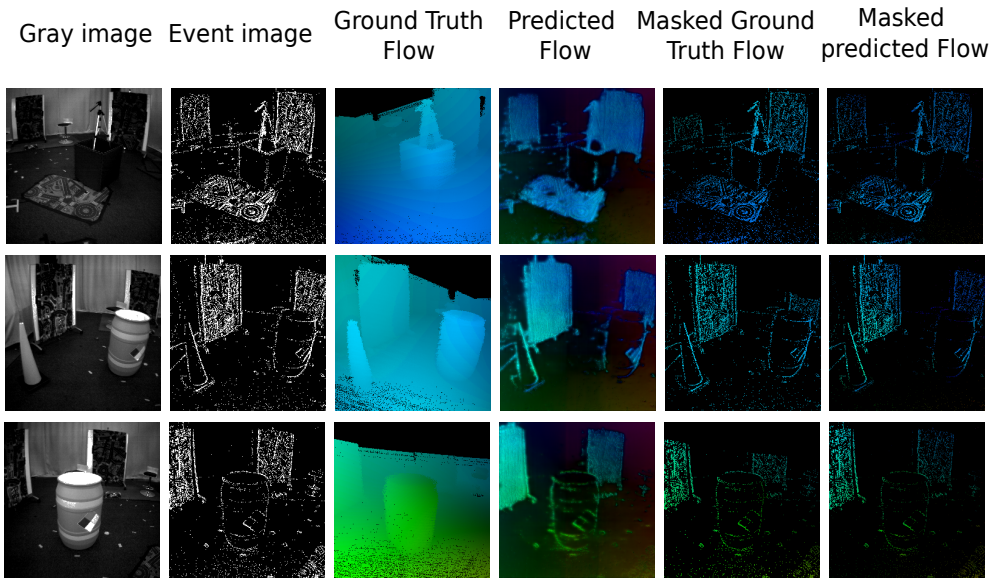
(a) outdoor day1



(b) indoor flying1



(c) indoor flying2



(d) indoor flying3

Figure 2.6 – Visualization of the optical flow estimation.

brightness changes, resulting in few optical flow predictions in the corresponding areas. The results show that 3D-FlowNet can predict optical flow accurately in both indoor and outdoor day1 sequences. Since we only trained our network on the outdoor sequence, this proves that the proposed 3D-FlowNet generalizes well to a variety of environments.

2.4 Discussion

This chapter introduces the optical flow estimation for event-based vision. We first present a model-based approach used in our object detection and tracking framework. Then we propose our deep learning-based approach. In contrast to previous works, we keep the 3D structure of the event data and use the 3D convolution module to process and squeeze it. The results show that our method can provide more accurate optical flow estimation compared to the state-of-the-art approaches, 13% improvement compared to the Spike-FlowNet [31], and 32% compared to the EV-FlowNet [92].

Compared to the model-based approach, deep learning-based approaches do not require complex modeling procedures. Thay can be easily adapted to different tasks, while the model-based approach still requires extra modeling processes. This advantage is shown in the following chapters; our model-based object detection and tracking framework require modeling all the problems, such as object detection and sensor fusion. But with the deep learning-based approach, we can quickly realize object detection by adding a detection head and sensor fusion by stacking the tensors. However, the model-based approach is easier to analyze the performance and problems of the system. So it is still worth developing the model-based approach. In the following two chapters, we will introduce our model-based and deep-learning-based approaches for object detection and tracking.

Chapter 3

Model-based Object Detection and Tracking with a fusion of Event-based Camera and Frame-based Camera

3.1 Introduction

After we successfully extract the motion information from the event data by calculating the optical flow, we can then use it for our object detection and tracking task. Object tracking is an essential task within the field of autonomous vehicles and it as been widely addressed in the computer vision community. The extensive use of high-powered computers, the availability of high-quality and inexpensive video cameras, and the increasing need for automated video analysis within autonomous vehicles applications, have generated significant interest in object tracking algorithms.

In the camera space, the tracking problem can be defined as estimating the trajectory of objects in the image plane as it moves around a scene. In other words, a tracker assigns consistent labels to the tracked objects in different video frames. Additionally, depending on the tracking domain, a tracker can also provide object-centric information, such as the orientation, speed, or type of the object. Tracking objects can be complex due to the limitation of the sensors and algorithms. During the tracking process, random numbers of objects can birth and die at any time. Crowded objects can increase the complexity of the data association. The occlusion problem will lead to wrong motion estimation and downgrade the performance of the tracking. Apart from the modeling problems, the noise from the camera will further increase the difficulty of tracking. The noise detection will create false tracks, and the missing detection could lose current tracks.

3.1.1 Related work

For event-based vision, most research focused on feature detection and tracking rather than object detection and tracking. Earlier methods to object tracking assumed a priori user-defined c ontours, w hich w as s hown i n s everal w orks [22], [46], [47]. F or instance, [46] forms event images in which an exponential function decays the value of each pixel with time as the parameter. The classic Iterative Closest Point (ICP) algorithm is then used to match a pre-defined contour to the event images and realize object detection and tracking. Those methods simplify the detection and tracking problem by integrating a priori knowledge into the system; however, they have very limited use cases.

More generic tracking was proposed by [57], [77] by adapting proven algorithms from conventional computer vision. [57] used Harris corner detection, and [77] used the traditional Kanade–Lucas–Tomasi feature tracker (KLT) onto the computed event images.

[74] developed a high-temporal tracking algorithm with a fusion of the event and frame cameras. They used principled arguments of event data generation to justify the choice of relevant features and proposed a pipeline to extract features from the frame images. Then, an event-based tracking algorithm is used for feature tracking.

For the multi object tracking, algorithms that are based on Random Finite Sets (RFS) have been popular recently [63]. The RFS theory is proposed to model the Multi-Object Tracking (MOT) problems; it is the mathematically simplest version of point process theory [40]. It provides a carefully constructed practitioners' toolbox of explicit, rigorous, systematic, and general procedures. Among the family of MOT algorithms based on RFS, the Poisson multi-Bernoulli mixture (PMBM) is the state-of-art MOT algorithm [18], which has better performance than the cardinalized probability density (CPHD) filters and the generalized labeled multi-Bernoulli (GLMB) [78], [80]. The other competitors extract appearances of the objects from the image and achieve better association performance [79]. But the extraction requires an extra model, which will increase the complexity of the tracking algorithm, especially with multiple sensors.

3.1.2 Contributions

This chapter's main contribution is an object tracking framework [71] that combines both Event-based camera and Frame-based camera information. This framework uses the event camera to add robustness in the frame camera in poor light conditions or fast motion environments. We present a novel density-based clustering algorithm, called Spatial-

Temporal-Flow-Density-Based Spatial Clustering of Applications with Noise (STF-DBSCAN) for the detection based on event data. We proposed a tracking approach that uses a hybridisation of the event-based camera and frame-based camera processing, and this is the first approach that uses the event camera to assist the frame-based camera in tracking objects. The presented object tracking algorithms have three components: detector, fusion strategy, and Poisson multi-Bernoulli mixture (PMBM) filter . The detector using the event camera data is based on a clustering algorithm that makes use of the position, timestamp, and optical flow of the event data (calculated with the model-based optical flow estimation approach described in the chapter 1). For the frame image, we adopt an deep learning model (RetinaNet, [34]) and obtain the bounding box as the detection. Since the form of detection from the two sensors are different, and redundant measurements can reduce the Signal-to-noise ratio (SNR), a fusion strategy is needed to unite the detection and remove the redundant measurements. This work has been published in [71]

3.2 Proposed Approach

3.2.1 Basic Concepts and Algorithm Overview

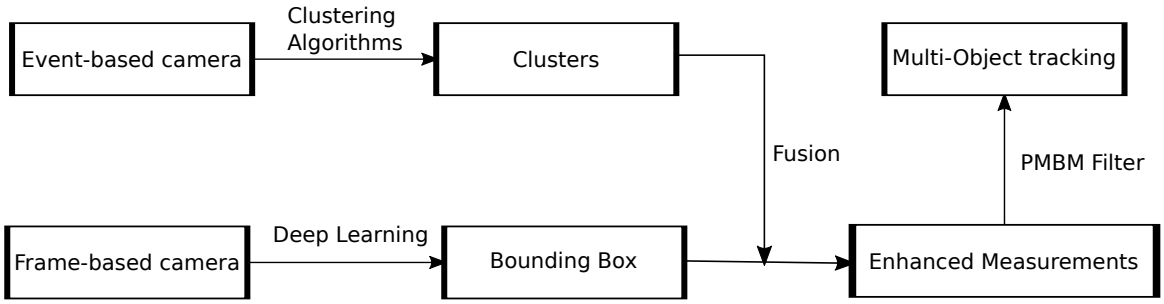


Figure 3.1 – Diagram of our approach.

Our framework aims to achieve more robust object tracking with a hybrid of event data and frame images. For the frame image, there are plenty of algorithms for reliable and stable object detection. But in the extreme lightning conditions, the quality of the frame image can be poor. The event camera can be used to fix the problem because of its high dynamic range. Figure (3.1) is the diagram of our approach, the clusters from the event camera and the detection (bounding box) from the frame camera are fused into the

final detection, and a PMBM filter is used for the final tracking.

The problem tackled in this paper is the processing of image sequence I_{t_k} and the groups of event data $\mathbf{e}_{t_k} = [e_1, e_2, ..., e_m]$ into a set of estimated tracks of objects $\hat{\mathbf{X}}_k = \{\boldsymbol{x}_{t_k}^1, \boldsymbol{x}_{t_k}^2, ..., \boldsymbol{x}_{t_k}^n\}$, as shown in Eq. (3.1),

$$\begin{cases} I_0, I_1, ..., I_{t_k} \\ \mathbf{e}_0, \mathbf{e}_1, ..., \mathbf{e}_{t_k} \end{cases} \quad \Longrightarrow \quad \hat{\mathbf{X}}_\mathbf{0}, \hat{\mathbf{X}}_\mathbf{1}, ...\hat{\mathbf{X}}_\mathbf{k} \tag{3.1}$$

here I_{t_k} denotes the frame image at time t_k, $\mathbf{e}_{t_k}$ represents a group of event data within the temporal window $[t_k - \Delta t, t_k + \Delta t]$ of time t_k. The object is represented by the state vector:

$$\boldsymbol{x} = [x, y, \dot{x}, \dot{y}] \tag{3.2}$$

where (x, y) is the location of the object, $(\dot{x}, \dot{y})$ is the speed of object. Correspondingly, the measurement vector is:

$$\boldsymbol{z} = [x_z, y_z, w_z, h_z, conf] \tag{3.3}$$

where (x_z, y_z) is the measured location of the object which is the top-left point of the bounding box, (w_z, h_z) is the width and height of the bounding box, and $conf$ is the confidence indicator of the measurement. The quantity $\mathbf{Z} = \{\boldsymbol{z}^1, \boldsymbol{z}^2, ..., \boldsymbol{z}^n\}$ is called the measure set.

For the frame image, the detection is in the form of a bounding box $\boldsymbol{b} = [x_1, y_1, w, h]$, and the top-left point $p = (x_1, y_1)$ is used as the object location. For the event-based camera, the detection is in the form of the clusters $c = \{e_1, e_2, ..., e_k\}$. There are two type of sensors in our framework, the fusion strategy is required to generate the final enhanced measurements set $\mathbf{Z}_k = \mathbf{Z}_k^c \uplus \mathbf{Z}_k^1 \uplus \cdots \uplus \mathbf{Z}_k^n$, where $\mathbf{Z}_k^c, \mathbf{Z}_k^1 \cdots \mathbf{Z}_k^n$ are independent sets and symbol $\uplus$ stands for disjoint union, represent that $\mathbf{Z}_c \cup \mathbf{Z}_1 \cup ... \cup \mathbf{Z}_n = \mathbf{Z}$ and $\mathbf{Z}_c, \mathbf{Z}_1, ..., \mathbf{Z}_n$ are mutually disjoint (and possibly empty). $\mathbf{Z}_k^c$ is the set of clutter measurements, $\mathbf{Z}_k^i$ is the set of measurements produced by target i at time k. After that, the PMBM filter will take measurement $\mathbf{Z}_k$ as input and output the tracks $\hat{\mathbf{X}}_k$. Next, we will introduce our object detection method with the event-based camera.

3.2.2 Object Detection

In this section, we describe how to generate object detection with event data. We first use the optical flow algorithm [1] describe in the previous chapter to generate optical flow,

and then we propose the STF-DBSCAN method to cluster the event data into detections based on the events and their corresponding optical flow. For the object detection with frame-based camera, there has been exponential growth with rapid development of new tools and techniques [86], so in our framework, we use RetinaNet to generate bounding box as detection, and the details are in [34].

ST-DBSCAN is an extension to DBSCAN for clustering spatial-temporal data [6], it uses two distance parameters (ϵ_s, ϵ_t) to measure the similarity of data according to their spatial and temporal attribute. The raw event data is just spatial-temporal data that can use the ST-DBSCAN. However, it is not suffgicient to only consider the spatial-temporal information. Suppose there are two objects that stand closely and move in a different direction simultaneously, the optical flow is different between the two objects but the ST-DBSCAN will consider the two objects as one object. Therefore, we propose the STF-DBSCAN, to measure the similarity of spatial, temporal and motion information.

In order to support three dimensions, we use $(\epsilon_s, \epsilon_t, \epsilon_{of})$ to measure the similarity of spatial, temporal and optical flow, respectively. For event data we have (x, y, u, v, t), which denotes the location, optical flow, and timestamp respectively. Two points will be considered as neighbours only when the three conditions are met at the same time:

$$\begin{aligned} &\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2)} < \epsilon_s \\ &|(t_1 - t_2)| < \epsilon_t \\ &\sqrt{(u_1 - u_2)^2 + (v_1 - v_2)^2} < \epsilon_{of} \end{aligned} \tag{3.4}$$

The algorithm of our STF-DBSCAN is shown in algorithm 1. The input of the algorithm is a set of event data and their optical flow: $\mathbf{e}_k = \{e_1, e_2, ..., e_m\}$, where $e_m = (x, y, t, u, v)$. (x, y) is the location of the event, t is the timestamp and (u, v) is the optical flow of the event. The parameters of the algorithm are $(\epsilon_s, \epsilon_t, \epsilon_{of}, minPts, \Delta\epsilon)$. The $minPts$ is the required minimum number of points that a new cluster can be created and the $\Delta\epsilon$ is the threshold value to merge a cluster. The output is the label for all the event points. First, the algorithm will pick one point and find all the neighbours of this point. If the number of neighbour points exceed the threshold $minPts$, this point is a core point and a cluster is formed. For the newly created cluster, we will expand it using a chain rules. If there is another core point in the cluster, we will try to merge the new cluster into the previous cluster as shown in lines 12-23 of Algorithm 1.

In the next section, we will introduce the fusion strategy between the detection of event

Algorithm 1 STF-DBSCAN

Input: $\mathbf{e}_k, \epsilon_s, \epsilon_t, \epsilon_{of}, minPts, \Delta\epsilon$
Output: labels

for e_m in $\mathbf{e}_k$ **do**
  if $label_m$ undefined **then**
    $Neighbors = find\_neighbor(e_m, \epsilon_s, \epsilon_t, \epsilon_{of})$
    if $|Neighbors| < minPts$ **then**
      $label_m = noise$
      continue
    end if
    $label_m \leftarrow$ next cluster label
    for p in $Neighbors$ **do**
      $label_p = label_m$
    end for
    $cluster =$ push($point$ in $Neighbors$)
    while $cluster$ is not empty **do**
      $point = cluster.pop()$
      $Neighbors_p = find\_neighbor(point, \epsilon_s, \epsilon_t, \epsilon_{of})$
      if $|Neighbors_p| > minPts$ **then**
        for q in $Neighbors_p$ **do**
          if ($label_q \neq noise$ or $label_q$) undefined and $distance(cluster_Avg, q) < \Delta\epsilon$ **then**
            $label_q = label_m$
          end if
        end for
      end if
    end while
  end if
end for

camera and frame camera, the details about the PMBM filter will also be discussed.

3.2.3 Fusion strategy

After obtaining the clusters, we propose a fusion strategy to hybrid the detection from the frame image and clusters from the event data. The fusion strategy is shown in Algorithm 2. The input are the bounding boxes $\mathcal{B}_k$ of the frame image and the clusters C_k from event data. The output is the enhanced measurement set $\mathbf{Z}_k$. First, we generate the bounding box $\mathcal{B}_{ek}$ for the clusters C_k. After that, we calculate the Intersection over Union (IoU) of each pair of the bounding boxes between the $\mathcal{B}_{ek}$ and $\mathcal{B}_k$. If the IoU is greater than the threshold ϵ, then the two bounding boxes are paired, and the two measurements are merged.

Algorithm 2 Fusion clusters and Detections

Input: $C_k, \mathcal{B}_k$
Output: Enhanced measurements $\mathbf{Z}_k$
Generate bounding box matrix $\mathcal{B}_{ek}$ for event clusters C_k
for $\boldsymbol{b}$ in $\mathcal{B}_k$, $\boldsymbol{b_e}$ in $\mathcal{B}_{ek}$ **do**
  if $IoU(\boldsymbol{b}, \boldsymbol{b_e}) \geq \epsilon$ **then**
    $\mathbf{Z}_k(\cdot) = [x, y, w, h, 2]$
    $\mathcal{B}_{ek}.pop(\boldsymbol{b_e})$
    $\mathcal{B}_k.pop(\boldsymbol{b})$
  end if
end for
for $\boldsymbol{b}$ in $\mathcal{B}_k$ **do**
  $\mathbf{Z}_k(\cdot) = [x, y, w, h, 1]$
end for
for $\boldsymbol{b_e}$ in $\mathcal{B}_{ek}$ **do**
  $\mathbf{Z}_k(\cdot) = [x, y, w, h, 0]$
end for

In order to reduce noise detection, the enhanced measurements need a label to indicate the confidence of detection. If the detection is from the frame camera and the event camera together, which means the detection from the two sensors are paired, then the measurement will be labeled with '2' considered as high confident detection. The detection of the frame camera will be labeled with '1', and the detection from the event camera is considered as low confident detection and marked with '0'. Next, we will discuss the details of the implementation of the PMBM filter.

3.2.4 PMBM filter

The PMBM filter [18] is a multi-object tracking algorithm based on the Bayesian filter framework with Random Finite Sets (RFS). In the standard framework for target tracking, we have a single target state $\boldsymbol{x} \in R^{n_x}$ (in our case $\boldsymbol{x} = [x, y, \dot{x}, \dot{y}, w, h]$ as defined before), and a multi-target state $\mathbf{X} \in \mathcal{F}(R^{n_x})$, where $\mathbf{X}$ is a set whose elements are single target state vectors and $\mathcal{F}(R^{n_x})$ denotes the space of all finite subsets of R^{n_x} . In the update step, the state is observed by measurements that are represented as a set $\mathbf{Z} \in R^{n_x}$. Given a prior multi-object density $f(\cdot)$ and the multi-object likelihood $l(\mathbf{Z}|\mathbf{X})$, the posterior multi-target density of X given the observation $\mathbf{Z}$ according to the Bayes' rule:

$$q(\mathbf{X}) = \frac{l(\mathbf{Z}|\mathbf{X})f(\mathbf{X})}{\rho(\mathbf{Z})} \tag{3.5}$$

where $\rho(\mathbf{Z})$ is the normalising constant is:

$$\begin{aligned} \rho(\mathbf{Z}) &= \int l(\mathbf{Z}|\mathbf{X})f(\mathbf{X})\delta\mathbf{X} \\ &\sum_{n=0}^{\infty} \frac{1}{n!} \int l(\mathbf{Z}|\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n) \times f(\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\})\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\} \end{aligned} \tag{3.6}$$

The Bayesian filtering recursion then can be obtained with the prediction step.

$$p(\mathbf{X}_{t+1}) = \int \kappa(\mathbf{X}_{t+1}|\mathbf{X}_t)q(\mathbf{X}_t)\delta\mathbf{X} \tag{3.7}$$

where $\mathbf{X}_{t+1} \in$ denotes the state at the next time step and $\kappa(\mathbf{X}_{t+1}|\mathbf{X}_t)$ is the transition density of the state given the state $\mathbf{X}$. Figure (3.2) is the diagram of the Bayesian filtering recursion. For each time step, we start with the posterior density of previous step $q(\mathbf{X}_t)$. After the prediction step, we obtain the prior density at the next time step $p(\mathbf{X}_{t+1})$. With the measurement $\mathbf{Z}_{t+1}$ and update step, the posterior density of next step $q(\mathbf{X}_{t+1})$ can be obtained. For each time step, the hypothesis that has biggest weight will be extracted as the track.

With the Bayesian equation, we still need to derive the likelihood $l(\mathbf{Z}|\mathbf{X})$ and multi-object density $f(\mathbf{X})$. For the likelihood $l(\mathbf{Z}|\mathbf{X})$ in the Bayesian filtering recursion: given the set $\mathbf{X}$ of targets, the set $\mathbf{Z}$ of measurements as defined before. The likelihood density can be written as:

$$l(\mathbf{Z}|\{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}) = e^{-\lambda_c} \sum_{\mathbf{Z}_c \biguplus \mathbf{Z}_1 \biguplus ... \biguplus \mathbf{Z}_n = \mathbf{Z}} [c(\cdot)]^{\mathbf{Z}^c} \prod_{i=1}^{n} \hat{l}(\mathbf{Z}_i|\boldsymbol{x}_i) \tag{3.8}$$

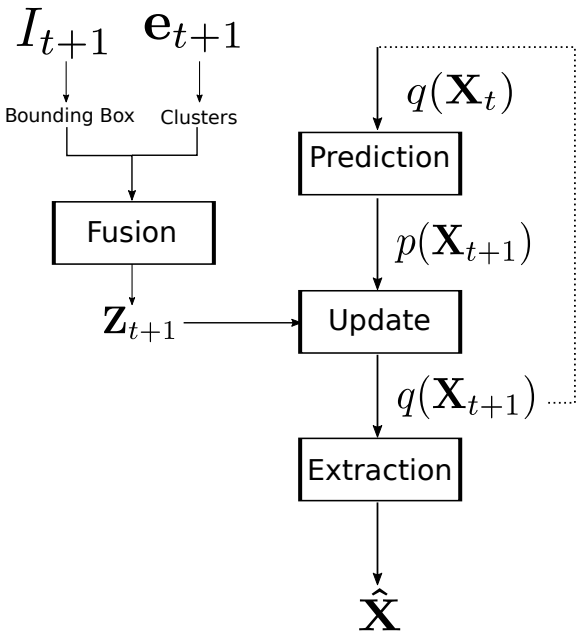


Figure 3.2 – Overview of the PMBM filter.

$$\hat{l}(\mathbf{Z}|\boldsymbol{x}) = \begin{cases} p_d \cdot p(\boldsymbol{z}|\boldsymbol{x}) & \mathbf{Z} = \{\boldsymbol{z}\} \\ 1 - p_d & \mathbf{Z} = \emptyset \\ 0 & |\mathbf{Z}| > 1 \end{cases} \tag{3.9}$$

where $l(\mathbf{Z}|\mathbf{X})$ is the density of measurement set $\mathbf{Z}$ given target $\mathbf{X}$ and $\hat{l}(\mathbf{Z}|\boldsymbol{x})$ is the density of measurement set $\mathbf{Z}$ given target $\boldsymbol{x}$; $\lambda_c = \int c(\boldsymbol{z})d\boldsymbol{z}$ and $[c(\cdot)]^{\mathbf{Z}} = \prod_{\boldsymbol{z}\in\mathbf{Z}} c(\mathbf{Z}), [c(\cdot)]^{\emptyset} = 1$. For the set of measurements produced by i_{th} object $\mathbf{Z}_i$, we get $\mathbf{Z}_i = \emptyset$ with probability $1 - p_d$, which corresponds to the case where the target is not detected, and $\mathbf{Z}_i = \{\boldsymbol{z}\}$ where $\boldsymbol{z}$ has a density $p(\boldsymbol{z}|\boldsymbol{x}_i)$ with probability p_d, which corresponds to the case where the target is detected. The equation (3.8) means that given the $\mathbf{Z}$, we go through all the possible sets $\mathbf{Z}_c, \mathbf{Z}_1, ..., \mathbf{Z}_n$ that meet the requirement $\mathbf{Z}_c \uplus \mathbf{Z}_1 \uplus ... \uplus \mathbf{Z}_n = \mathbf{Z}$. In other words, each term of the sum considers a measurement-to-object association hypothesis. Figure (3.3) is an example of the association hypothesis when there are two measurements $\{\boldsymbol{z}_1, \boldsymbol{z}_2\}$ and $n = 1$. Note that the hypothesis that assigns multiple measurements to one object will have zero probability. So the first hypothesis will be removed from the sum.

Due to the computation limitations, we still need to develop the approximation of the multi-object density. The PMBM filter is the state-of-the-art approximation, which uses

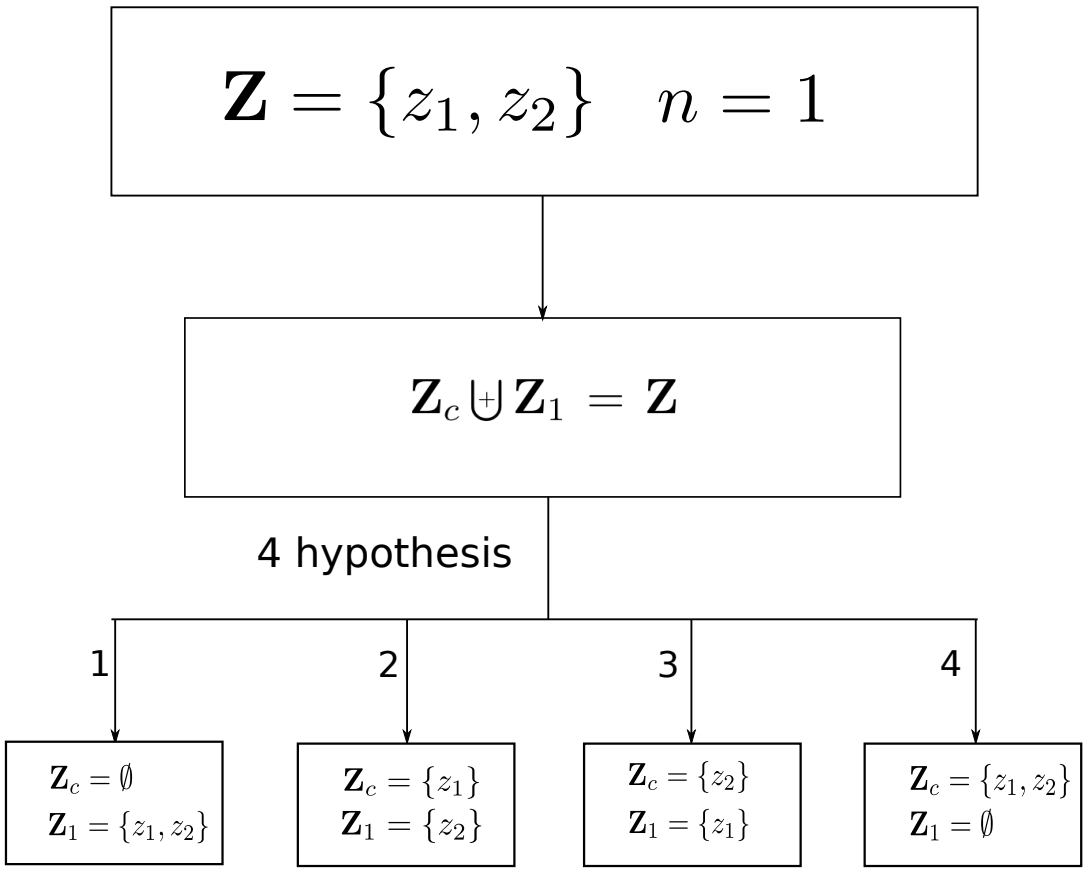


Figure 3.3 – Example of the association hypothesis.

the Poisson distribution representing the birth of targets and the multi-Bernoulli mixture representing the tracks. The PMBM density can be defined as:

$$f(\mathbf{X}) = \sum_{\mathbf{Y} \biguplus \mathbf{W} = \mathbf{X}} f^p(\mathbf{Y}) f^{mbm}(\mathbf{W}) \tag{3.10}$$

where $f^p(\cdot)$ is a Poisson density and $f^{mbm}(\cdot)$ is a multi-Bernoulli mixture. The Poisson $f^p(\cdot)$can be written as:

$$f^p(\mathbf{X}) = e^{-\int \lambda(x)dx} [\lambda(\cdot)]^{\mathbf{X}} \tag{3.11}$$

where the $\lambda(\cdot)$ is the birth intensity, $[\lambda(\cdot)]^X$ is set power, defined as $[\lambda(\cdot)]^{\mathbf{X}} = 1$ if $\mathbf{X} = \emptyset$ and $[\lambda(\cdot)]^{\mathbf{X}} = \prod_{\boldsymbol{x} \in \mathbf{X}} \lambda(\boldsymbol{x})$ if $\mathbf{X} \neq \emptyset$.

The multi-Bernoulli mixture has multiplicative weights is defined as:

$$f^{mbm}(\mathbf{X}) \propto \sum_{j} \sum_{\mathbf{X}_1 \biguplus \cdots \biguplus \mathbf{X}_n} \prod_{i=1}^{n} w_{j,i} f_{j,i}(\mathbf{X}_i) \tag{3.12}$$

where $\propto$ stands for proportionality, j is an index over all global hypotheses (components of the mixtures), n is the number of potentially detected targets and, $w_{j,i}$ and $f_{j,i}(\cdot)$ are the weight and the Bernoulli density of potentially detected target i under the j_{th} global hypothesis. The Bernoulli densities have the expression:

$$f_{j,i} = \begin{cases} 1 - r_{j,i} & \mathbf{X} = \emptyset \\ r_{j,i} p_{j,i}(x) & \mathbf{X} = \{\boldsymbol{x}\} \\ 0 & otherwise \end{cases} \tag{3.13}$$

where $r_{j,i}$ is the probability of existence and $p_{j,i}(\cdot)$ is the state density given that it exists. If there is only one mixture component in the multi-Bernoulli mixture in equation 3.12 (j = 1), we can obtain a multi-Bernoulli density:

$$f^{mb}(\mathbf{X}) = \sum_{\mathbf{X}_1 \biguplus \cdots \biguplus \mathbf{X}_n} \prod_{i=1}^{n} f_{1,i}(\mathbf{X}_i) \tag{3.14}$$

This derivation indicates that a new Bernoulli component should be created for each new measurement, where its existence corresponds to the possibility that the measurement is the first detection of a new target, and non-existence corresponds to the case that the measurement is a false alarm or it corresponded to a different, previously detected object. Also, we assume one object can create at maximum one measurement, so the number of potentially detected objects corresponds to the number of measurements up to the current time. The weight of global hypothesis j is proportional to the product of the hypothesis weights $\prod_{i=1}^{n} w_{j,i}$ for the n potentially detected targets. If a potentially detected target i is not considered in the global hypothesis j, which means that its originating measurement was assigned to another object, the $w_{j,i} = 1$ and the probability of existence of $f_{j,i}(\cdot)$ is zero.

With the Eq. (3.10) and (3.12), we can have:

$$f(\mathbf{X}) \propto \sum_{\mathbf{Y} \biguplus \mathbf{X}_1 \biguplus \cdots \biguplus \mathbf{X}_n = \mathbf{X}} f^{p}(\mathbf{Y}) \sum_{j} \prod_{i=1}^{n} w_{j,i} f_{j,i}(\mathbf{X}_i) \tag{3.15}$$

Note that, given $\mathbf{X}$, $\mathbf{X}_i$ can be either empty or a single element set (otherwise, its density $f_{j,i}(\cdot)$ is zero) and Y can have any cardinality within the constraint $\mathbf{Y} \uplus \mathbf{X}_1 \uplus \cdots \uplus \mathbf{X}_n = \mathbf{X}$.

After we obtain the approximation of the multi-object density, we still to give the definition of the transition density and likelihood density in the Bayesian filtering recursion.

$$\kappa(\mathbf{X}_{t+1}|\mathbf{X}_t) = \begin{cases} 1 & \mathbf{X}_{t+1} = \emptyset, \mathbf{X}_t = \emptyset \\ 1 - p_s & \mathbf{X}_{t+1} = \emptyset, \mathbf{X}_t = \{\boldsymbol{x}_t\} \\ p_s \times \pi(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t) & \mathbf{X}_{t+1} = \{\boldsymbol{x}_{t+1}\}, \mathbf{X}_t = \{\boldsymbol{x}_t\} \\ 0 & otherwise \end{cases} \tag{3.16}$$

Here, p_s is the survival possibility and $\pi(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t)$ is the single object motion model. According to the Eq. (3.16), if the target "dies" ($\mathbf{X}_{t+1} = \emptyset, \mathbf{X}_t = \{\boldsymbol{x}_t\}$), then the entire trajectory will no longer be a member of the set of current trajectories. If the target survives, then the trajectory is extended by one time step. For the single object motion model, we use constant velocity motion model. Given the object state described, we can define the motion model as:

$$\begin{aligned} x_t &= x_{t-1} + dt \cdot \dot{x}_{t-1} \\ y_t &= y_{t-1} + dt \cdot \dot{y}_{t-1} \\ \dot{x}_t &= \dot{x}_{t-1} \\ \dot{y}_t &= \dot{y}_{t-1} \end{aligned} \tag{3.17}$$

Note that we use the Poisson distribution representing the birth target and the multi-Bernoulli mixture representing the tracks. So, for the likelihood density, we first give the update for a Poison prior using the likelihood in Eq. (3.8), For $\mathbf{Z} = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_m\}$, we can write the likelihood (5) as:

$$l(\{\boldsymbol{z}_1, ..., \boldsymbol{z}_m\}|\mathbf{X}) = e^{-\lambda_c} \sum_{\mathbf{U} \uplus \mathbf{Y}_1 \uplus \cdots \uplus \mathbf{Y}_m = \mathbf{X}} [1 - p_d]^{\mathbf{U}} \prod_{i=1}^{m} \hat{l}(\boldsymbol{z}_i|\mathbf{Y}_i) \tag{3.18}$$

$$\hat{l}(\boldsymbol{z}|\mathbf{Y}) = \begin{cases} p_d \times p(\boldsymbol{z}|\boldsymbol{y}) & \mathbf{Y} = \{\boldsymbol{y}\} \\ c(\boldsymbol{z}) & \mathbf{Y} = \emptyset \\ 0 & |\mathbf{Y}| > 1 \end{cases} \tag{3.19}$$

Eq. (3.18) means that we decompose the set $\mathbf{X}$ of targets into all possible sets $\mathbf{U}, \mathbf{Y}_1 ... \mathbf{Y}_m$

with the constraint that $\mathbf{U} \uplus \mathbf{Y}_1 \uplus ... \uplus \mathbf{Y}_m = \mathbf{X}$. $\mathbf{U}$ represents the undetected targets and $\mathbf{Y}_i$ represents the object that generate i_{th} measurement, which can be a single-element set containing the state of the object that gave rise to the measurement, or an empty set if the measurement is clutter. This is a different but equivalent way of expressing the data association hypotheses considered in Eq. (3.8).

For the likelihood representation for the for the Bernoulli component, we have

$$l_b(\mathbf{Z}|\mathbf{Y}, \mathbf{X}_1, \cdots, \mathbf{X}_n) = \sum_{\mathbf{Z}_y \uplus \mathbf{Z}_1 \uplus ... \uplus \mathbf{Z}_n = \mathbf{Z}} l(\mathbf{Z}_y|\mathbf{Y}) \times \prod_{i=1}^{n} t(\mathbf{Z}_i|\mathbf{X}_i) \tag{3.20}$$

$$t(\mathbf{Z}_i|\mathbf{X}_i) = \begin{cases} p_d \times l(\boldsymbol{z}|\boldsymbol{x}) & \mathbf{X}_i = \{\boldsymbol{x}\}, \mathbf{Z}_i = \{\boldsymbol{z}\} \\ 1 - p_d & \mathbf{X}_i = \{\boldsymbol{x}\}, \mathbf{Z}_i = \emptyset \\ 1 & \mathbf{X}_i = \emptyset, \mathbf{Z}_i = \emptyset \\ 0 & otherwise \end{cases} \tag{3.21}$$

where $\mathbf{Z}_y$ represents both measurements from targets in $\mathbf{Y}$ and clutter, and $t(\mathbf{Z}_i|\mathbf{X}_i)$ is the likelihood for a set with zero or one measurement elements without clutter.

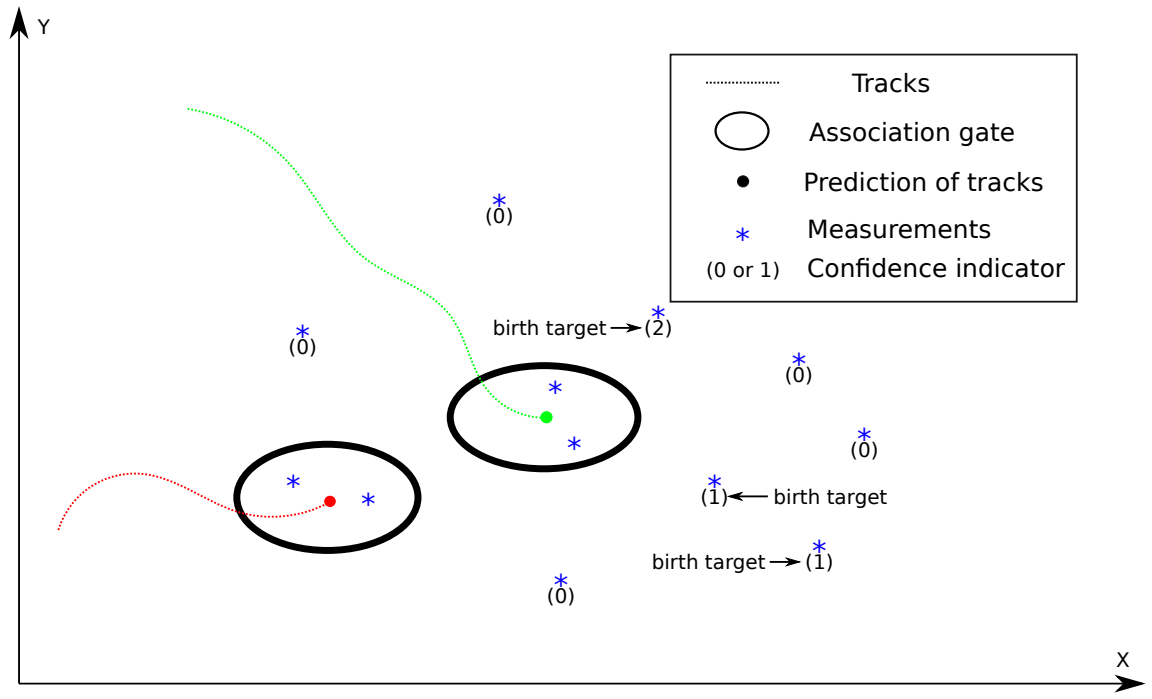


Figure 3.4 – The birth strategy of our approach.

At last, we still need to initialize the birth density because the Poisson density requires prior knowledge of the birth target location. Usually, all measurements that fail to associate with the current tracks will be considered birth objects. However, in our framework, the detection is from two sensors. And the detection from the event data can be very noisy. So we adopt the adaptive birth intensity strategy by using the confidence indicator in the measurement. Only the measurement that has a positive confidence indicator can be seen as a birth target. The strategy is shown in figure (3.4). For those measurements that fail to be associated with current tracks, only the one that has a positive confidence indicator can be used as a birth target.

3.3 Experiments

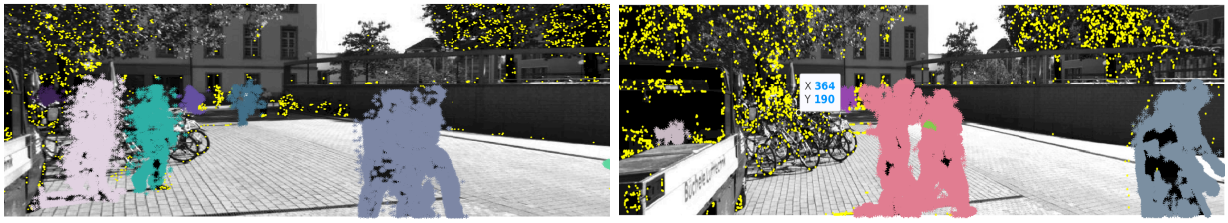
The evaluation contains two parts: detection and tracking. We first test our clustering algorithm on the MOT Challenge15 (MOT15) dataset [29]. We convert the dataset into the event-based camera version using the event camera simulator [58]. Then, we compare our algorithm with the traditional clustering algorithms and the state-of-the-art methods on three event-based camera datasets. The tracking performance is also evaluated on the converted MOT Challenge15 (MOT15) dataset and compared with the approach that only relies on the frame-based camera. At last, we test the approach with our experiment vehicle and DAVIS346 camera to examine the feasibility of our approach in the real environments.

3.3.1 Detection Rate

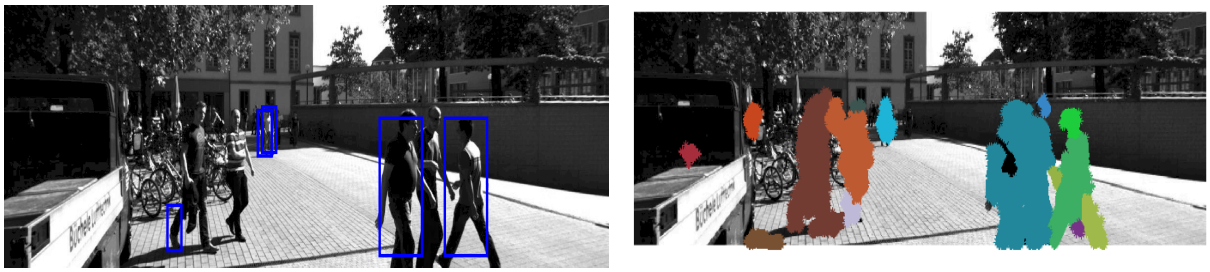
For the clustering algorithm, the detection rate is used to evaluate the detection performance [65]. We first test our approach with the converted MOT15 dataset. Then we test our approach with three challenging event-based camera datasets: EED [43], MOD [60], EV-IMO [44], where EED is a real dataset in extreme light conditions; MOD is a synthetic dataset designed for training the neural network of object detection; EV-IMO is a real dataset focusing on the fast camera motion and rich texture surface.

The quantitative results are shown in table (3.1). The k-means and DBSCAN represent the two cluster algorithms that use the position information only. DAViS Flow + k-means represents the k-means algorithm that uses optical flow and position information. For the three challenging event-based camera datasets, the results show that our approach has a

(a) Miss Detection of frame image



(b) Detection of event data



(c) Detection of the dense pedestrians

Figure 3.5 – Detection evaluation based on the event data.

better performance than the other clustering algorithms. Compare to the [43], [65], our approach has similar performance. But our approach requires fewer accumulated events and is naturally synchronized with the frame image. So, our detection method is more suitable for hybridizing the event camera and the frame camera. Our approach's performance in extreme light scenarios (EED) and the MOT15 is good. But for the scenario that is full of texture and has fast camera motion (MOD, EV-IMO), our approach still needs to be improved. The performance is unsatisfactory under two circumstances: 1. fast self-motion of the event-based camera. 2. the object has no visible movement relative to the camera. The qualitative results are shown in Figure (3.5). The potential solution is to calculate the global motion of the camera using the IMU data and filter out the event point caused by self-motion. Figure (3.5) (a) is the detection of the frame image and de-

Table 3.1 – Comparison of the Detection Rate

Methods	Detection rate ↑ for dataset (%)			
	MOT15	EED	MOD	EV-IMO
k-means	64.34	61.46	36.83	42.59
DBSCAN	79.78	80.91	40.15	45.37
DAViS Flow + k-means	74.05	85.49	46.83	**55.73**
Mitrokhin et al. [43]	-	88.93	**70.12**	48.79
Stoffregen et al. [65]	-	**93.17**	-	-
Ours	**80.15**	92.32	64.36	48.82

tection is shown in blue bounding boxes. Figure (3.5) (b) is the clustering results of our approach. The clusters are represented in different colors and the yellow points are those event data that belong to the background. The results show that when the detection from frame camera is missing, the event data can provide backup detection.

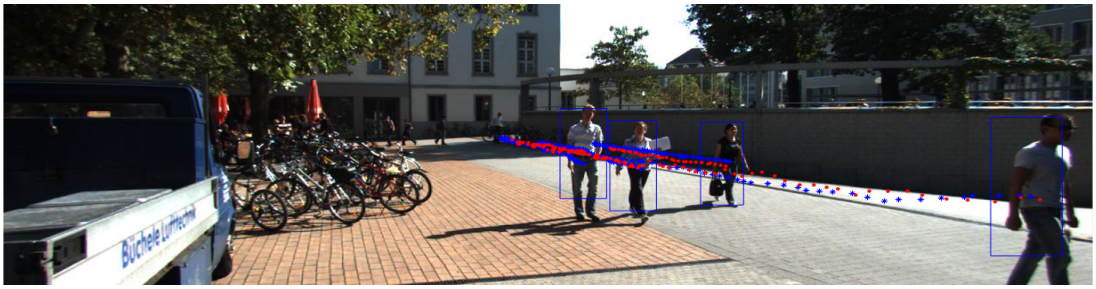
3.3.2 Multi-object Tracking Performance

For the object tracking evaluation, we compare our approach with the PMBM filter that only uses the detection of frame images. We use the Higher Order Tracking Accuracy (HOTA) metric described in [38]. This metric balances the effect of performing accurate detection, localization, and association into a single unified metric: HOTA. The quantitative results of our approach are shown in the table (3.2). The results show that our approach has better overall performance then tracking with frame camera only. But the association accuracy (AssA) and detection recall (DetRe) of our approach is slightly worse, because the event camera increases the noise of the detection.

Table 3.2 – Comparison of Object tracking results

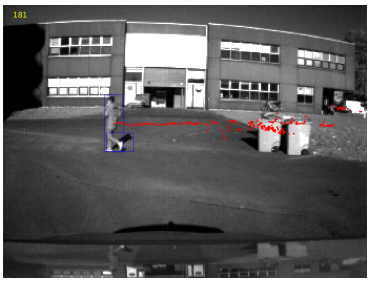
Methods	HOTA	DetA	AssA	DetRe	DetPr	AssRe	AssPr	LocA
Our approach	**31.21**	**48.54**	20.39	**64.82**	54.63	**21.92**	**73.45**	**75.40**
frame image only	28.96	41.72	**20.57**	48.49	**60.22**	21.44	72.98	74.90

The evaluation we have performed, is based on the event camera simulator, all the data is synthetic by doing interpolation between two frame images. Their spatial resolution is different from the real event camera. In order to examine that our approach is also feasible with the real event camera and vehicle, we test it on our experiment vehicle. Figure (3.6) is the tracking results of a simple scenario: the experiment vehicle is moving forwards

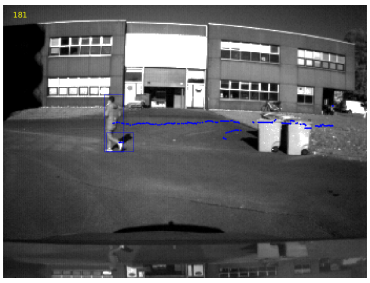


(a) Track visualization of our approach

(b) Autonomous vehicle



(c) Our approach



(d) Without event camera

Figure 3.6 – Visualization of our tracking approach. (a) is the visualization of the estimation and ground truth tracks for KITTI-17, the red point is the estimated trajectory, and the blue star is the ground truth. (b)-(d) Visualization of our tracking approach in the real environment. The person is running from right to left in front of the vehicle, the tracking with frame camera is in blue and the track of our approach is red.

slowly while a pedestrian is running across the road. The camera we use is the DAVIS346. Figure (3.6) (c) is the track of the person based on our approach, figure (3.6) (d) is the track based on the PMBM filter and frame camera. When the pedestrian passes the trash bin, the object is lost due to the interfere of the trash bin. But our approach fixes this problem with the detection from the event-based camera.

3.4 Discussion

In this chapter, we presented a novel approach that realizes object tracking based on a hybrid of event camera and frame camera. This approach is the first framework that tracks the object using the hybrid of the event camera and frame camera. The approach exploits the temporal information by calculating the optical flow with the event data and generating clusters according to their position, location, and optical flow. Then the clusters are combined with detection from the frame image and fed into the PMBM filter. Our

approach utilizes the advantages of the event camera to achieve better tracking performance by providing complementary detection for the frame camera. Because a clustering algorithm is adopted, our method has limitations in the crowded object and rich texture background environment. In the future, improving the object detection algorithm for the event camera under rich texture and crowded environment is a perspective direction. We will also need to record datasets for the scenario with motion blur and extreme light conditions where the frame camera fails, so we can develop a more robust detection algorithm with the datasets.

As shown in our approach, the modeling process is complicated, and the whole framework is computationally inefficient; the performance still needs to be improved. So, in the next chapter, we will present a deep-learning-based approach that can realize object detection and sensor fusion in one model and reduce the system's complexity.

Chapter 4

Deep learning-based Moving Object Detection

4.1 Introduction

As discussed in the previous chapter, the model-based approach could be complex and inefficient. So we proposed a deep-learning-based approach to improving the performance of our framework. And since the event cameras can provide rich temporal information (motion information), we modified our task to moving object detection, which is more complex but essential to the autonomous driving system.

An Autonomous Vehicle (AV) needs an accurate perception of its surrounding environment to work reliably and safely. Its perception system should transform raw sensory data such as image pixels into semantic information for scene understanding [2]. For an autonomous vehicle, it is required to fully estimate the motion model of each of the surrounding participants and to plan the ego-trajectories based on their future states to reduce collision risks. There are two main classes of motion in a typical autonomous driving scene: The surrounding moving objects and the scene motion generated by the ego vehicle. Due to the ego vehicle's motion and constraints related to the image formation, it is very challenging to classify the surrounding objects as moving or static because even static objects will be perceived as moving. Motion segmentation implies that the two tasks have to be performed jointly. The first focuses on object segmentation, in which objects of specific interesting classes are highlighted, such as pedestrians or vehicles. The second focuses on motion classification, in which a classifier predicts whether the observed object is moving or static.

For the segmentation task, the frame camera can achieve good performance with the appearance information of the frame images. However, the frame images are not enough for the motion classification task. Although we can use two consecutive images to simulate the motion information, but they are not accurate enough for the complicated motion

classification task. The event camera cannot provide appearance information but can produce precise timestamps that contain rich motion information. So the frame camera and event camera are good complementary for the motion segmentation task.

4.1.1 Related work

Because the event-based cameras can produce large numbers of events, especially when the camera is moving, identifying the events produced by objects of interest in the general event stream is difficult. For autonomous vehicles, the important events are those produced by moving traffic participants and not those produced by the self-motion of the vehicle. In this case, motion segmentation of event data is required.

[21] proposed a method to detect and track circular objects (such as a ball) in the presence of clutter caused by camera self-motion. This work used an adapted directed Hough transform, where each incoming event contributes to values in Hough space on possible radii of a target circle. To reduce the complexity of the task and to improve the robustness, a directed Hough transform is proposed with the estimated optical flow. The assumption is that most events are generated at the circumference of a circular object, such as a ball. [22] extend this work using a particle filter, and improve the robustness of the tracking. A drawback of these methods is that they require a priori knowledge of the target shape.

[76] proposed a method that detects corners in the event stream from an event camera mounted on a robot. The method consists of two stages: the prior learning stage and the operation stage. In the prior learning stage, they learn to estimate the corner's motion as a function of the robot joint velocities in a static scene. In the operation stage, corners are detected and clustered in the event stream; if there is a discrepancy between these corners' motions and the expected motions, those events are segmented out. While able to detect more universal objects, this method depends on the pre-learned setting and configuration and only works for scenes and objects with corners.

[43] proposed a method that estimates the camera's motion with focus optimization and considers it the dominant motion. The average timestamp of the motion-compensated event image is then computed, and outliers from the dominant motion can be detected. These outliers are assumed to be independently moving objects, and the resulting segmentation is applied to surrounding events by using flood-fill in the event image. The disadvantage is that segmentation in densely textured environments and crowded moving objects fails.

[65] proposed an improved method by providing a segmentation framework for arbitrary motion models and numbers of moving objects. This method uses focus optimization on multiple motion models at the same time; a probabilistic model is used for each event belonging to each model. The motion parameters and the event probabilities are then updated iteratively in an Expectation Maximisation (EM) approach.

For the sensor fusion, former works mainly focused on the fusion between LiDAR and the frame-based camera. [39] proposed an algorithm for 3D semantic segmentation using LiDAR and camera. [56] proposed a semantic segmentation algorithm using fusion between images and optical flow. In [55], a Convolutional Neural Network (CNN) architecture is proposed for Moving Object Detection under low-light conditions by capturing motion information from both camera and LiDAR sensors. It also presented the KittiMoSeg dataset, which provides the moving object mask for the KITTI sequence.

4.1.2 Contributions

The main contribution of this chapter is to propose a novel Deep Neural Network architecture for moving objects detection. This approach realized moving object detection and sensor fusion in one model, reducing the complex modeling process and increasing the computation efficiency. As mentioned previously, traditional frame-based cameras cannot provide temporal information, and the event-based cameras lack appearance information and spatial consistency. The two features are both important for moving object detection. We address this problem by proposing a fusion network model that can use the information from the two sensors simultaneously and achieve better performances. We train and evaluate the proposed EV-FuseMODNet network using the extended KittiMoSeg dataset [55]. The results show that the proposed approach outperforms current state-of-the-art methods, by achieving 27.5% improvement compared to the FuseMODNet[55], and 36.7% compared to the MODNet[64], this work has been submitted in IEEE Intelligent Vehicles Symposium 2023.

4.2 Proposed Approach

In this section, we first explain the proposed network structure, including three feature encoders, the fusion structure, and the decoder. We will also discuss our encoding method for the event data because they are asynchronous and cannot be directly fed into the

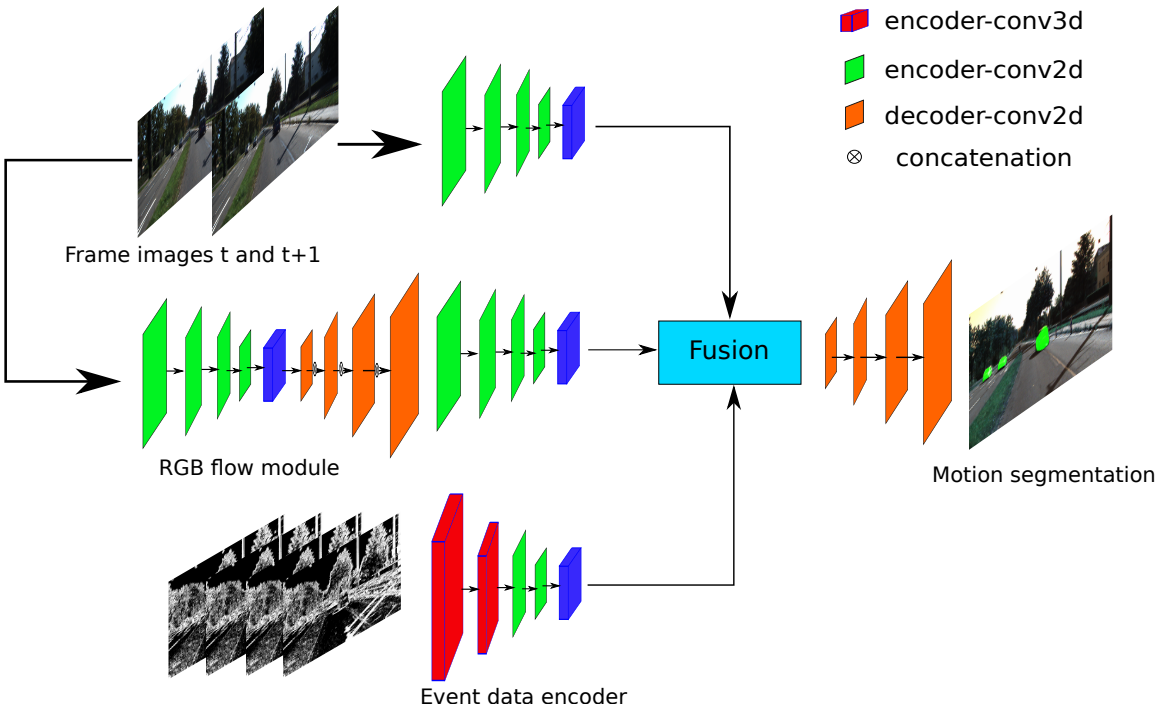


Figure 4.1 – Network structure of the ev-FuseMODNet

neural network.

4.2.1 Network structure

The overall architecture of the proposed EV-FuseMODNet, is shown in Figure (4.1). It contains three main parts: frame-based optical flow, frame-based image processing, and event data processing.

For the frame-based optical flow, we directly adopt the RAFT [75] model since it achieves the state of the art performance and high efficiency in inference time. A 2D-encoder is then used after the optical flow estimation to extract features for the fusion step, shown as the RGB flow module in figure 4.1. This module is designed to provide precise motion information to the model, and only the 2D-encoder will be trained during the training process. The details about the 2D-encoder are shown in the figure (4.2). Green rectangle denote the 2D convolutional block, where [2DConv,a,b,c,(d_1, d_2), e] denotes a 2D convolutional layer with the kernel size a, stride b, output channel c, padding size (d_1, d_2) and normalization layer e (BN is Batch Normalization). Blue rectangle is the ResBlock, and its structure is shown in figure (4.3). The brown rectangle means ReLU activation function.

For the frame image processing, we also use a 2D-encoder structure to extract the features as shown at the top of figure (4.1). This 2D-encoder has the same design as the 2D-encoder used for the RGB flow estimation, as shown in the figure (4.2). Two consecutive images will be sent into the encoder so it can provide the appearance and general motion features. Note that this encoder is pretrained with the cityscape segmentation dataset.

We adopt the 3D-encoder (as shown in 2.3) [69] for event data processing. This choice is linked to the fact that it can better preserve the spatial-temporal information of the event data compared to the 2D-encoder. The details of the 3D-encoder are shown in figure (4.4). Red rectangle denotes the 3D convolutional modules, where [3DConv,a,b,c,(d_1, d_2, d_3), e] denotes a 2D convolutional layer with the kernel size a, stride b, output channel c, padding size (d_1, d_2, d_3) and normalization layer e. The green and blue rectangles represent the 2D convolutional block and ResBlock, respectively. Since the event data is asynchronous and cannot be used in the network directly, an encoding method is also required here. Given a set of N input events $E_N = (x_i, y_i, t_i, p_i), i \in [1, N]$, and a time depth D to discretize the time dimension of event data, we accumulate each group of event data into images as defined in Eq. (2.28).

Here, (x, y) denotes the position of the event, p is the polarity of the event, and δ

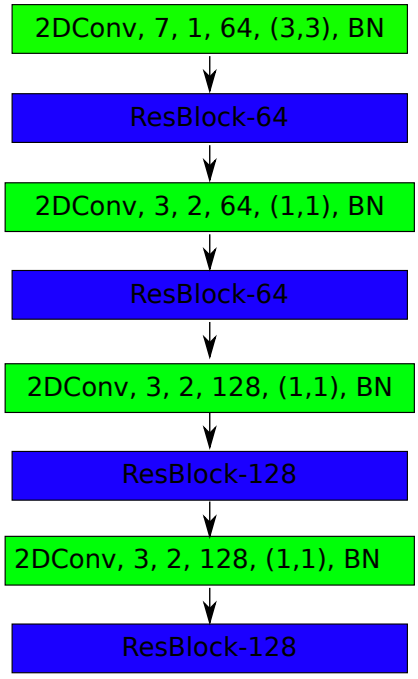


Figure 4.2 – The details of the 2D-encoder for frame image processing and RGB flow processing

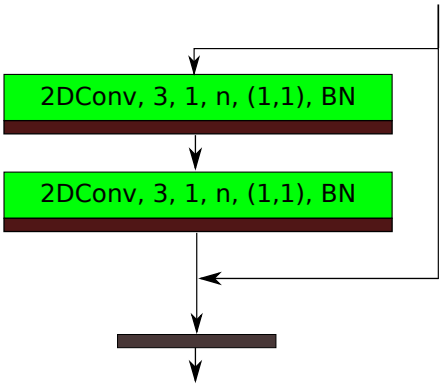


Figure 4.3 – The details of the ResBlock

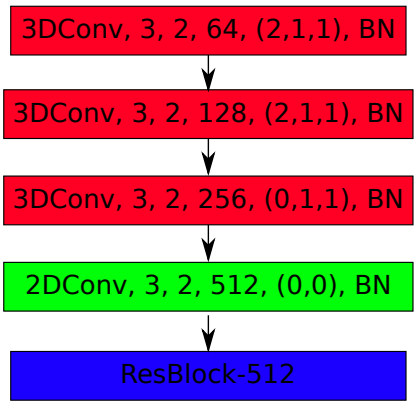


Figure 4.4 – The details of the 3D-encoder for the event data processing

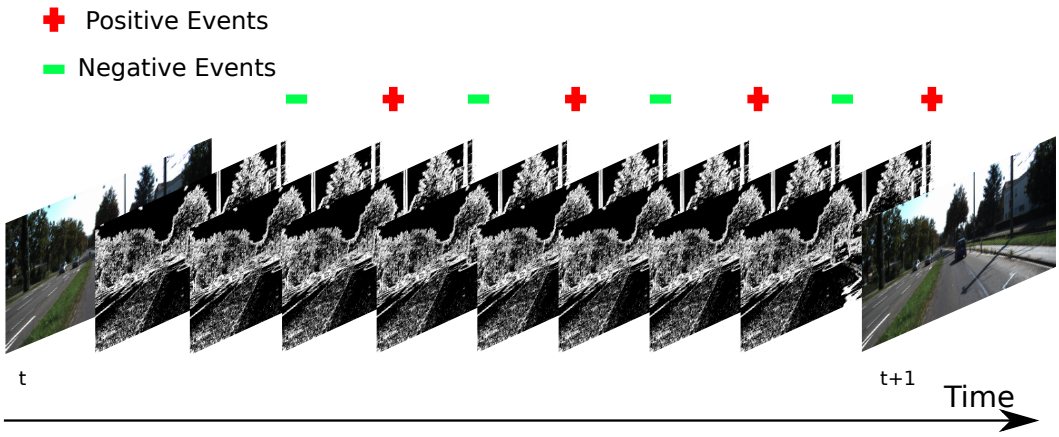


Figure 4.5 – Visualization of our event encoding representation.

is the Kronecker delta operator. $k_b(\cdot)$ denotes bi-linear sampling kernel. The generated event image I is a $(2, D, H, W)$ tensor, where the number 2 represents the positive and negative polarity, D is the discretized time depth, and (H, W) are respectively the height and width of the image. Fig. 4.5 is an example of the event data representation where $D = 4$.

4.2.2 The fusion architecture

The fusion architecture of the EV-FuseMODNet is shown in figure (4.6). A mid fusion strategy is adopted in our model. Mid-Fusion represents feature-level-fusion where features are extracted from each input separately using an encoder that is exclusive

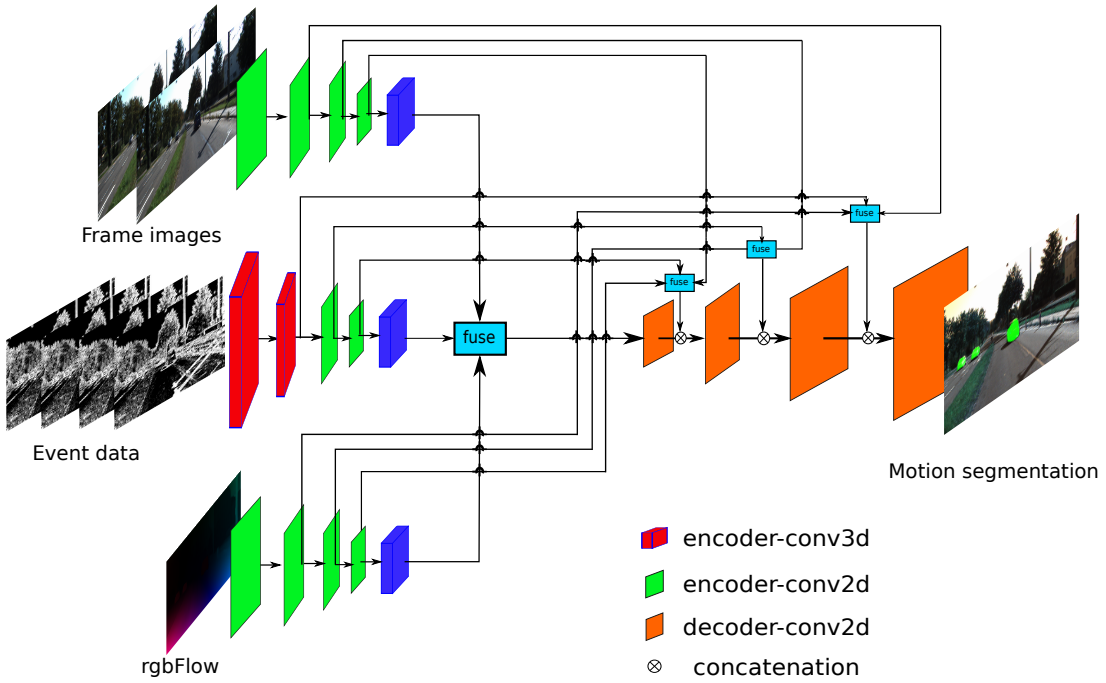


Figure 4.6 – Fusion details of the ev-FuseMODNet

to each input. The fusion is done by concatenating feature maps generated from each stream before feeding them into the decoder. The details about the decoder is shown in the figure (4.7). Orange rectangle denote the transpose 2D convolutional block, where [2DConvT,a,b,c,(d_1, d_2), e] denotes a 2D transpose convolutional layer with the kernel size a, stride b, output channel c, padding size (d_1, d_2) and normalization layer e. The green and blue rectangles are 2D convolutional block and ResBlock, respectively. There is a skip connection from each encoder to the corresponding decoder. For the skip connection between 2D-encoder and the decoder, the activation of the encoder is directly concatenated with the activation of the decoder. For the skip connection of the 3D-encoder for the event data processing, the 3D activation $(C \times D \times W \times H)$ is flattened into a 2D tensor $((C \cdot D) \times W \times H)$ first, then it is concatenated with the activation of the decoder.

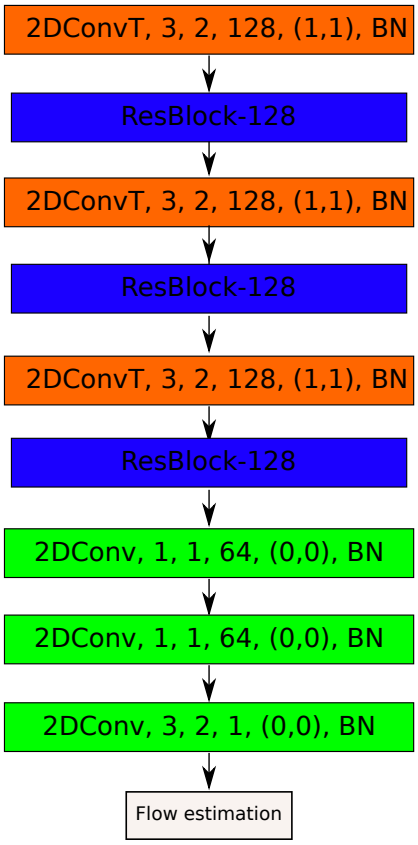


Figure 4.7 – The details of the dencoder of the ev-FuseMODNet

4.3 Experiments

4.3.1 Dataset and Implementation Details

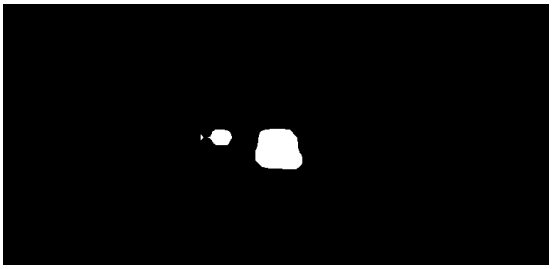
We trained all proposed models end-to-end with weighted binary cross-entropy loss for 100 epochs and batch size of 8. The Adam optimizer is used with a learning rate of 1e-5 and a weight decay rate of 5e-4. We evaluate our methods with the extended KittiMoSeg dataset. The extended KittiMoSeg provides the motion mask for the raw KITTI sequence and contains 12919 images.

Our work is to develop a complete system for moving object detection that can work under poor illumination conditions. For that purpose, we must evaluate our approach with challenging low-illumination scenes where frame cameras would fail. As far as we know, there exists no dataset providing low-illumination or night scenes in addition to the information needed for the moving object detection task. So the night images are generated using the Image-toImage translation technique [36]. The generated dark image is used to simulated the nighttime as shown in figure (4.8).

4.3.2 Results

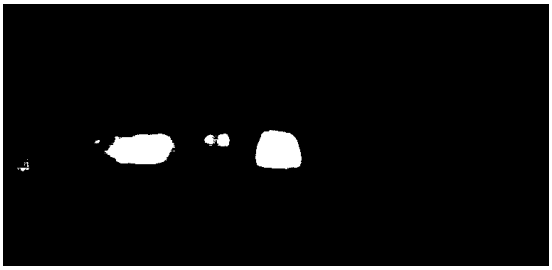
Table (4.1) shows the results of the Intersection-over-Union (IoU) for the moving objects in comparison to previous moving object detection approaches [55], [64]. The results show that our approach shows a 22.7% improvement in the daytime KITTI sequence and a 31.2% improvement in the generated nighttime KITTI sequence. We attribute the improvement to the fusion with event data. Compared to the LiDAR, the event-based camera is a better complementary to the frame-based camera for the moving object detection task. The frame-based camera transmits images at a fixed rate. So it provides the appearance information but no motion knowledge. On the opposite, the event-based camera monitors the brightness change of each pixel and provides the precise timestamp value of each event data. So it lacks appearance information but can give more precise motion features. Also, the improvement in the Dark-KITTI sequence also proves that the event-based camera can improve the robustness of the model in a bad light environment.

Figure (4.8) is the qualitative result of our approach. Results show the benefit of fusion, where the network was able to segment the moving objects in both daytime and nighttime scenes. Note that the ground truth mask is imperfect because it misses the left vehicle. According to the rendering image, our approach performs better since it recognizes the left-



(a) Ground truth of the motion mask

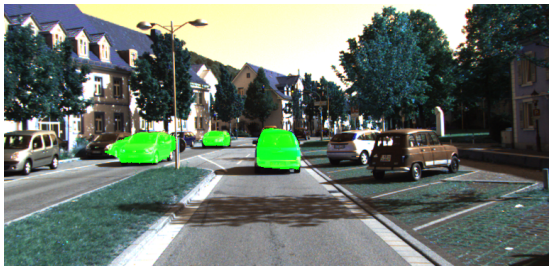
(b) Accumulated event visualization



(c) Predicted motion mask in the nighttime

(d) Rendering nighttime image

(e) Predicted motion mask in the daytime



(f) Rendering daytime image

(g) rgbFlow of the daytime image

(h) rgbFlow of the nighttime image

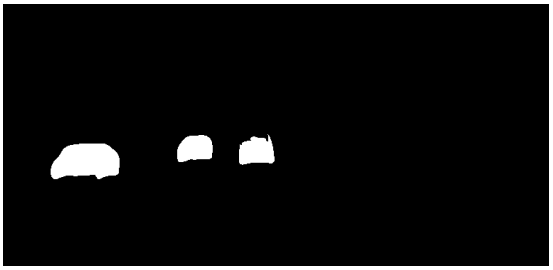
Figure 4.8 – Qualitative result of the ev-FuseMODNet

Table 4.1 – Quantitative assessment of our approach compared to the state-of-the-art methods

Methods	Moving IoU
KITTI	
RGB-only	32.7
RGB+rgbFlow[64]	49.36
RGB+LidarFlow	41.64
RGB+rgbFlow+LidarFlow[55]	51.46
Ours-RGB+rgbFlow+Events	**63.16**
Dark-KITTI	
RGB-only	26.5
RGB+rgbFlow[64]	39.5
RGB+LidarFlow	38.5
RGB+rgbFlow+LidarFlow[55]	43.5
Ours-RGB+rgbFlow+Events	**57.51**

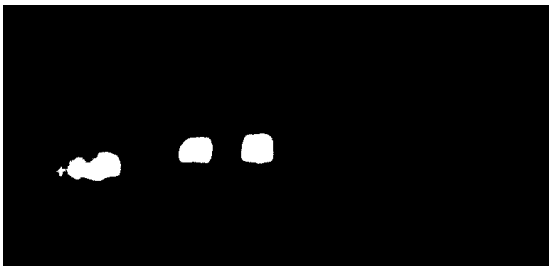
moving vehicle. However, the performance of our model in the nighttime is downgraded; Fig. 4.8 (c) shows that there is noise around the left vehicles, and the shape of the middle vehicle is not satisfactory. This is because the quality of the frame images downgrades under the low-illumination environment.

Figure (4.9) shows an example of failure of the ev-FuseMODNet. In this sample, the ego vehicle is moving forward, and the target moving objects are two opposite-direction cars and one same-direction car. In the daytime, our approach can provide an accurate estimation of moving objects because the frame camera provides enough appearance features to the model. However, during the nighttime, the prediction for the left car is not satisfying. Their two main reasons: the left car is the most dark area of the RGB image, which makes the frame camera completely fails. Also, the left car is moving beside the trees and grass. This creates a textured background and downgrades the performance of the event camera. we can verify this from the accumulated event visualization, the left car is almost invisible because of the noise background, but the contour of the rest car is still clear.

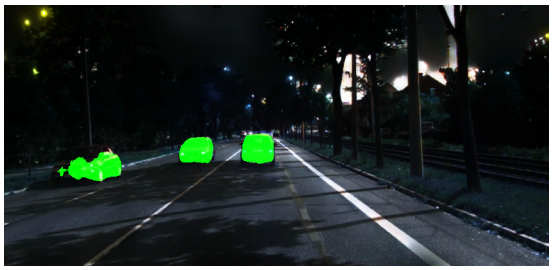


(a) Ground truth of the motion mask

(b) Accumulated event visualization

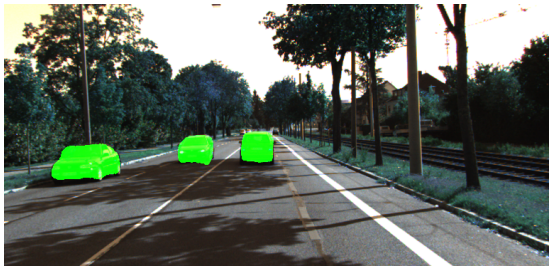


(c) Predicted motion mask in the nighttime



(d) Rendering nighttime image

(e) Predicted motion mask in the daytime



(f) Rendering daytime image

Figure 4.9 – Failure case of the ev-FuseMODNet

4.4 Conclusions

In this chapter, we propose EV-FuseMODNet, a deep fusion neural network for moving object detection, using a fusion of the frame-based camera and event-based camera data. This architecture fuses appearance features and motion information that is captured from both frame-based cameras and event-based cameras. The results show that our approach can generate more accurate (27.5%-36.7%) moving object segmentation due to the fusion of the event-based camera. However, the performance of this moving object detection model still needs to be improved, so a more efficient and robust model is still needed.

In the future, adding more sensors (e.g. LiDAR) to the fusion model is a perspective direction; better fusion architecture is also required since more sensors will increase the computation time.

Conclusion

Event-based cameras are a novel, bio-inspired type of visual sensor. Their generated data shifted from synchronous, conventional absolute intensity images to asynchronous update, low-latency events. By reporting only relative changes in brightness, event-based cameras have several advantages compared to frame-based cameras: low latency (timestamped with a 1MHz clock), low power usage (on the order of mW), and high dynamic range ($\approx$ 140 dB). Event data are reported in the format of a tuple, consisting of position (x, y), the polarity of brightness change p, and time t, which provides each event with a fine-grained timestamp. Due to their working principle, event-based cameras emit samples at precisely the time of change of brightness, limited only by the Contrast Threshold (the brightness change threshold required to trigger an event) and refractory period (the period after firing during which a pixel can not emits a new event).

Because event-based cameras respond only to the dynamics of the scene, they are a natural fit for algorithms that describe motion, such as optical flow estimation, object tracking, or motion segmentation. However, because event-based cameras do not encode spatial patterns and relationships in the intuitive way that frame-based cameras do, many of the methods that are used in frame-based vision cannot be directly applied to event-based vision. Also, because the event data do not provide the appearance information, fusing it with the frame cameras is better to achieve more robust performance. This thesis first tries to solve this problem by proposing a model-based object-tracking framework that includes optical flow calculation, object detection, sensor fusion, and object tracking. Every module of the model-based framework is independent, so the whole framework is large and computationally inefficient. So, deep learning-based approaches are proposed for optical flow estimation, sensor fusion, and motion segmentation.

Chapter 1 presents the optical flow estimation with the event data. We discuss this first because this is the basis for the signal processing of event cameras. We introduced the model-based optical flow that we used in our model-based object detection and tracking framework. We also proposed a deep learning-based optical flow estimation model, which uses 3D convolution to preserve the temporal information of the event data and achieve better performance. Key contributions of this work are:

(i) a 3D deep learning-based encoder for the event data;

(ii) an optical flow estimation model for event cameras. We showed that this approach performed better than previous Lucas-Kanade [37] based methods [59] and deep learning-based methods [31], [92].

Chapter 2 presents an object detection and tracking framework with a fusion of the event camera and frame camera. This framework contains five modules: Object detection algorithm for the frame images, optical flow estimation with the event data, object detection using the optical flow and event data, detection fusion, and object tracking. In this framework, the event-based camera provides clusters with the optical flow as detection; the frame camera provides the bounding box as the detection. The two types of detections are fused according to their intersection over union (IoU). The random finite set theory-based-PMBM filter is used for object tracking. Key contributions of this work were:

(i) an object detection algorithm for the event cameras;

(ii) a fusion strategy for the event-based and frame-based camera;

(iii) a framework that realizes object tracking with a fusion of event-based and frame-based camera.

Chapter 3 presents a deep learning-based sensor fusion and moving object detection approach. In this work, we integrated sensor fusion and moving object detection in an end-to-end approach. This approach uses precise temporal information from the event data and rich appearance information from the frame images to achieve better performance compared to the state-of-the-art methods. Key contributions of this work are: (i) a moving object detection algorithm for the event cameras; (ii) a deep learning-based fusion strategy for the event camera and frame camera; (iii) the investigation of the advantage of the event-based camera about the high dynamic range using the generated dark KITTI sequence.

This research has achieved good results for optical flow estimation and motion segmentation from event-based cameras. Still, we see future works for additional research. For the model-based object detection and tracking framework, two significant challenges remain to be solved.

The first one is the model-based object detection algorithm. The current clustering algorithm cannot perform well under a texture environment, especially when the self-motion is fast. Also, the clustering algorithm will detect all objects, whether moving or static. The potential way is to introduce the IMU information and distinguish the

event data between the foreground and background before the clustering algorithm. The foreground represents the segmentation caused by the moving object, and the background is caused by the self-motion. By introducing the IMU information, we could obtain the self-motion data and remove all the event data that is irrelevant to the moving object.

The second challenge is the fusion strategy; the current matching strategy only depends on the Intersection over Union (IoU). This means that we only use the location information for the fusion, which is insufficient. One promising way is to use the feature point descriptors, such as Scale-invariant feature transform (SIFT) and Features from accelerated segment test (FAST). Both descriptors have their event version, so it is possible to develop the relationship between the event-based descriptors and the frame-based descriptors.

The third challenge is the deep-learning-based fusion and moving object detection approach. The dataset we are using is designed for the frame-based cameras and Lidar. We used the event camera simulator to generate the data, but it could be better to record the data with the event camera. The challenging scenarios, such as extreme illumination environment and fast motion, should be considered when preparing the dataset. Although powerful results can be achieved using traditional neural networks by processing events in batches, this methodology can introduce unwanted additional latency. We believe that an exciting new research direction could be asynchronous Spiking Neural Networks (SNNs), which is a natural fit with the event data.

Our research shows that event data contains rich temporal information that can produce high-quality optic flow and motion segmentation when combined with frame cameras. This work positions event-based cameras as a valuable tool in mobile robotics, from autonomous vehicles to drones. With the recent advancements in computer vision, deep learning, and robotics capabilities, this is an exciting time to explore the possibilities of this new frontier of computer vision. This thesis serves as an important foundation for future research on sensor fusion and motion segmentation.

BIBLIOGRAPHY

[1] M. Almatrafi and K. Hirakawa, DAViS Camera Optical Flow, *IEEE Transactions on Computational Imaging*, vol. 6, 2020, 2020.

[2] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, A Survey on 3D Object Detection Methods for Autonomous Driving Applications, *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, 2019, pages: 3782–3795, 2019.

[3] R. Benosman, S.-H. Ieng, C. Clercq, C. Bartolozzi, and M. Srinivasan, Asynchronous frameless event-based optical flow, *Neural networks : the official journal of the International Neural Network Society*, vol. 27, Nov. 2011, pages: 32–7, Nov. 2011.

[4] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, Event-Based Visual Flow, *IEEE Transactions on Neural Networks*, vol. pp, Nov. 2013, pages: 1, Nov. 2013.

[5] J. Binas, D. Neil, S.-C. Liu, and T. Delbruck, DDD17: End-To-End DAVIS Driving Dataset, Nov. 2017, Nov. 2017.

[6] D. Birant and A. Kut, ST-DBSCAN: An algorithm for clustering spatial–temporal data, *Data & Knowledge Engineering*, vol. 60, no. 1, 2007, pages: 208–221, 2007, Intelligent Data Mining.

[7] C. Brandli, R. Berner, M. Yang, S.-C. Liu, and T. Delbruck, A 240 × 180 130 dB 3 µs Latency Global Shutter Spatiotemporal Vision Sensor, *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, 2014, pages: 2333–2341, 2014.

[8] T. Brosch, S. Tschechne, and H. Neumann, On event-based optical flow detection, *Frontiers in Neuroscience*, vol. 9, 2015, 2015. [Online]. Available: `https://www.frontiersin.org/articles/10.3389/fnins.2015.00137`.

[9] J. Carneiro, S.-H. Ieng, C. Posch, and R. Benosman, Event-based 3D reconstruction from neuromorphic retinas, *Neural Networks*, vol. 45, 2013, pages: 27–38, 2013.

[10] T. Delbruck, Frame-free dynamic digital vision, *Proceedings of the International Symposium on Secure-Life Electronics, Advanced Electronics for Quality Life and Society*, Mar. 2008, Mar. 2008.

[11] T. Delbruck and M. Lang, Robotic goalie with 3 ms reaction time at 4% CPU load using event-based dynamic vision sensor, *Frontiers in Neuroscience*, vol. 7, 2013, pages: 223, 2013. [Online]. Available: `https://www.frontiersin.org/article/10.3389/fnins.2013.00223`.

[12] T. Delbruck, Neuromorophic vision sensing and processing, in *Proceedings of the 2016 46th European Solid-State Device Research Conference (ESSDERC)*, 2016, pages: 7–14.

[13] P. Diehl and M. eCook, Unsupervised Learning of Digit Recognition Using Spike-Timing-Dependent Plasticity, *Frontiers in Computational Neuroscience*, vol. 9, Aug. 2015, Aug. 2015.

[14] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, FlowNet: Learning Optical Flow With Convolutional Networks, in *Proceedings of the Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015.

[15] M. Faisal and J. Barron, High Accuracy Optical Flow Method Based on a Theory for Warping: Implementation and Qualitative/Quantitative Evaluation, Aug. 2007, pages: 513–525.

[16] T. Finateu, A. Niwa, D. Matolin, K. Tsuchimoto, A. Mascheroni, E. Reynaud, P. Mostafalu, F. Brady, L. Chotard, F. LeGoff, H. Takahashi, H. Wakabayashi, Y. Oike, and C. Posch, 5.10 A 1280×720 Back-Illuminated Stacked Temporal Contrast Event-Based Vision Sensor with 4.86µm Pixels, 1.066GEPS Readout, Programmable Event-Rate Controller and Compressive Data-Formatting Pipeline, in *Proceedings of the 2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, 2020, pages: 112–114.

[17] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, *et al.*, Event-based vision: A survey, *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 1, 2020, pages: 154–180, 2020.

[18] A. F. Garcia-Fernandez, J. L. Williams, K. Granström, and L. Svensson, Poisson Multi-Bernoulli Mixture Filter: Direct Derivation and Implementation, *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, 2018, pages: 1883–1901, 2018.

[19] D. Gehrig, M. Gehrig, J. Hidalgo-Carrió, and D. Scaramuzza, Video to Events: Recycling Video Datasets for Event Cameras, in *Proceedings of the IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*, Jun. 2020.

[20] M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza, DSEC: A Stereo Event Camera Dataset for Driving Scenarios, *IEEE Robotics and Automation Letters*, 2021, 2021.

[21] A. Glover and C. Bartolozzi, Event-driven ball detection and gaze fixation in clutter, in *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pages: 2203–2208.

[22] ——, Robust visual tracking with a freely-moving event camera, in *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pages: 3769–3776.

[23] M. A. Goodrich, B. S. Morse, D. Gerhardt, J. L. Cooper, M. Quigley, J. A. Adams, and C. Humphrey, Supporting wilderness search and rescue using a camera-equipped mini UAV, *Journal of Field Robotics*, vol. 25, no. 1-2, 2008, pages: 89–110, 2008.

[24] B. Horn and B. Schunck, Determining Optical Flow, *Artificial Intelligence*, vol. 17, Aug. 1981, pages: 185–203, Aug. 1981.

[25] Y. Hu, J. Binas, D. Neil, S.-C. Liu, and Delbruck, DDD20 End-to-End Event Camera Driving Dataset: Fusing Frames and Events with Deep Learning for Improved Steering Prediction, in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, Nov. 2020.

[26] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017. [Online]. Available: `http://lmb.informatik.uni-freiburg.de//Publications/2017/IMKDB17`.

[27] S. Jain, B. Xiong, and K. Grauman, FusionSeg: Learning to Combine Motion and Appearance for Fully Automatic Segmentation of Generic Objects in Videos, Jul. 2017, pages: 2117–2126.

[28] C. Koch and I. Segev, *Methods in neuronal modeling: from ions to networks*. MIT press, 1998.

[29] L. Leal-Taixé, A. Milan, I. Reid, and S. Roth, MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking, *arXiv:1504.01942*, vol. abs/1504.01942, Apr. 2015, Apr. 2015.

[30] J. Lee, T. Delbruck, and M. Pfeiffer, Training Deep Spiking Neural Networks Using Backpropagation, *Frontiers in Neuroscience*, vol. 08, 2016, 2016. [Online]. Available: `https://www.frontiersin.org/articles/10.3389/fnins.2016.00508/full`.

[31] C. Lee, A. Kosta, A. Z. Zhu, K. Chaney, K. Daniilidis, and K. Roy, Spike-FlowNet: Event-based Optical Flow Estimation with Energy-Efficient Hybrid Neural Networks, in *Proceedings of the European Conference on Computer Vision*, Springer, 2020, pages: 366–382.

[32] C. Lee, S. S. Sarwar, P. Panda, G. Srinivasan, and K. Roy, Enabling Spike-based Backpropagation for Training Deep Neural Network Architectures, *Frontiers in Neuroscience*, vol. 14, 2020, pages: 119, 2020. [Online]. Available: `https://www.frontiersin.org/article/10.3389/fnins.2020.00119`.

[33] P. Lichtsteiner, C. Posch, and T. Delbruck, A 128× 128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor, *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, 2008, pages: 566–576, 2008.

[34] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, Focal Loss for Dense Object Detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, 2020, pages: 318–327, 2020.

[35] M. Litzenberger, B. Kohn, A. Belbachir, N. Donath, G. Gritsch, H. Garn, C. Posch, and S. Schraml, Estimation of Vehicle Speed Based on Asynchronous Data from a Silicon Retina Optical Sensor, in *Proceedings of the 2006 IEEE Intelligent Transportation Systems Conference*, 2006, pages: 653–658.

[36] M.-Y. Liu, T. Breuel, and J. Kautz, Unsupervised Image-to-Image Translation Networks, Mar. 2017, Mar. 2017.

[37] B. Lucas and T. Kanade, An Iterative Image Registration Technique with an Application to Stereo Vision (IJCAI), vol. 81, Apr. 1981.

[38] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, HOTA: A Higher Order Metric for Evaluating Multi-Object Tracking, *International Journal of Computer Vision*, 2020, pages: 1–31, 2020.

[39] K. Madawy, H. Rashed, A. Sallab, O. Nasr, H. Kamel, and S. Yogamani, RGB and LiDAR fusion based 3D Semantic Segmentation for Autonomous Driving, Jun. 2019.

[40] R. P. Mahler, *Advances in Statistical Multisource-multitarget Information Fusion*. Boston: ARTECH House, 2014.

[41] A. Maqueda, A. Loquercio, G. Gallego, N. Garcia, and D. 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 (CVPR)*, Jun. 2018, pages: 5419–5427.

[42] S. Meister, J. Hur, and S. Roth, UnFlow: Unsupervised Learning of Optical Flow with a Bidirectional Census Loss, in *Proceedings of the AAAI*, New Orleans, Louisiana, Feb. 2018.

[43] A. Mitrokhin, C. Fermüller, C. Parameshwara, and Y. Aloimonos, Event-Based Moving Object Detection and Tracking, in *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pages: 1–9.

[44] A. Mitrokhin, C. Ye, C. Fermüller, Y. Aloimonos, and T. Delbruck, EV-IMO: Motion Segmentation Dataset and Learning Pipeline for Event Cameras, in *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pages: 6105–6112.

[45] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, The Event-Camera Dataset and Simulator: Event-based Data for Pose Estimation, Visual Odometry, and SLAM, *The International Journal of Robotics Research*, vol. 36, Nov. 2016, Nov. 2016.

[46] Z. Ni, A. Bolopion, J. Agnus, R. Benosman, and S. Regnier, Asynchronous Event-Based Visual Shape Tracking for Stable Haptic Feedback in Microrobotics, *IEEE Transactions on Robotics*, vol. 28, no. 5, 2012, pages: 1081–1089, 2012.

[47] Z. Ni, S.-H. Ieng, C. Posch, S. Régnier, and R. Benosman, Visual Tracking Using Neuromorphic Asynchronous Event-Based Cameras, *Neural Computation*, vol. 27, no. 4, 2015, pages: 925–953, 2015.

[48] G. Orchard, R. Benosman, R. Etienne-Cummings, and N. V. Thakor, A spiking neural network architecture for visual motion estimation, in *Proceedings of the 2013 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2013, pages: 298–301.

[49] Z. Pan, J. Polden, N. Larkin, S. Van Duin, and J. Norrish, Recent progress on programming methods for industrial robots, *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, 2012, pages: 87–94, 2012.

[50] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert, Highly Accurate Optic Flow Computation with Theoretically Justified Warping, *International Journal of Computer Vision*, 2006, 2006.

[51] F. Paredes-Valles, K. Y. W. Scheper, and G. C. H. E. 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*, vol. 42, no. 8, 2020, pages: 2051–2064, 2020.

[52] F. Paredes-Vallés and G. C. de Croon, Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy, in *Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pages: 3446–3455.

[53] A. Ranjan and M. Black, Optical Flow Estimation Using a Spatial Pyramid Network, Jul. 2017, pages: 2720–2729.

[54] H. Rashed, A. El Sallab, S. Yogamani, and M. ElHelw, Motion and Depth Augmented Semantic Segmentation for Autonomous Navigation, in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019, pages: 364–370.

[55] H. Rashed, M. I. Ramzy, V. Vaquero, A. E. Sallab, G. Sistu, and S. K. Yogamani, FuseMODNet: Real-Time Camera and LiDAR Based Moving Object Detection for Robust Low-Light Autonomous Driving, *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pages: 2393–2402, 2019.

[56] H. Rashed, S. Yogamani, A. El-Sallab, P. Krizek, and M. El-Helw, Optical flow augmented semantic segmentation networks for automated driving, *arXiv preprint arXiv:1901.07355*, 2019, 2019.

[57] H. Rebecq, T. Horstschaefer, and D. Scaramuzza, Real-time Visual-Inertial Odometry for Event Cameras using Keyframe-based Nonlinear Optimization, Sep. 2017.

[58] H. Rebecq, D. Gehrig, and D. Scaramuzza, ESIM: an Open Event Camera Simulator, *Conf. on Robotics Learning (CoRL)*, Oct. 2018, Oct. 2018.

[59] B. Rueckauer and T. Delbruck, Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor, *Frontiers in neuroscience*, vol. 10, 2016, pages: 176, 2016.

[60] N. Sanket, C. Parameshwara, C. Singh, A. V. Kuruttukulam, C. Fermüller, D. Scaramuzza, and Y. Aloimonos, EVDodgeNet: Deep Dynamic Obstacle Dodging with Event Cameras, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pages: 10651–10657.

[61] C. Scheerlinck, H. Rebecq, T. Stoffregen, N. Barnes, R. Mahony, and D. Scaramuzza, CED: Color Event Camera Dataset, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Apr. 2019.

[62] C. Scheerlinck, H. Rebecq, D. Gehrig, N. Barnes, R. Mahony, and D. Scaramuzza, Fast image reconstruction with an event camera, in *Proceedings of the Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pages: 156–163.

[63] S. Scheidegger, J. Benjaminsson, E. Rosenberg, A. Krishnan, and K. Granström, Mono-Camera 3D Multi-Object Tracking Using Deep Learning Detections and PMBM Filtering, Jun. 2018, pages: 433–440.

[64] M. Siam, H. Mahgoub, M. Zahran, S. Yogamani, M. Jagersand, and A. El-Sallab, MODNet: Motion and Appearance based Moving Object Detection Network for Autonomous Driving, in *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pages: 2859–2864.

[65] T. Stoffregen, G. Gallego, T. Drummond, L. Kleeman, and D. Scaramuzza, Event-Based Motion Segmentation by Motion Compensation, Oct. 2019, pages: 7243–7252.

[66] T. Stoffregen, C. Scheerlinck, D. Scaramuzza, T. Drummond, N. Barnes, L. Kleeman, and R. Mahony, How to Train Your Event Camera Neural Network, Mar. 2020.

[67] D. Sun, S. Roth, and M. J. Black, Secrets of optical flow estimation and their principles, in *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pages: 2432–2439.

[68] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume, 2018.

[69] H. Sun, M.-Q. Dao, and V. Fremont, 3D-FlowNet: Event-based optical flow estimation with 3D representation, in *Proceedings of the 2022 IEEE Intelligent Vehicles Symposium, IV 2022, Aachen, Germany, June 4-9, 2022*, IEEE, 2022.

[70] H. Sun, M. Dao, and V. Frémont, 3D-FlowNet: Event-based optical flow estimation with 3D representation, in *Proceedings of the 2022 IEEE Intelligent Vehicles Symposium, IV 2022, Aachen, Germany, June 4-9, 2022*, IEEE, 2022, pages: 1845–1850. [Online]. Available: `https://doi.org/10.1109/IV51971.2022.9827380`.

[71] H. Sun and V. Fremont, Object Tracking with a Fusion of Event-Based Camera and Frame-Based Camera, in Sep. 2022, pages: 250–264.

[72] A. Taherkhani, A. Belatreche, Y. Li, G. Cosma, L. P. Maguire, and T. McGinnity, A review of learning in biologically plausible spiking neural networks, *Neural Networks*, vol. 122, 2020, pages: 253–272, 2020. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0893608019303181`.

[73] G. Tang, A. Shah, and K. P. Michmizos, Spiking neural network on neuromorphic hardware for energy-efficient unidimensional slam, in *Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2019, pages: 4176–4181.

[74] D. Tedaldi, G. Gallego, E. Mueggler, and D. Scaramuzza, Feature detection and tracking with the dynamic and active-pixel vision sensor (DAVIS), in *Proceedings of the 2016 Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)*, 2016, pages: 1–7.

[75] Z. Teed and J. Deng, Raft: Recurrent all-pairs field transforms for optical flow, in *Proceedings of the European conference on computer vision*, Springer, 2020, pages: 402–419.

[76] V. Vasco, A. Glover, E. Mueggler, D. Scaramuzza, L. Natale, and C. Bartolozzi, Independent motion detection with event-driven cameras, Jul. 2017.

[77] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High Speed Scenarios, in *Proceedings of the IEEE Robotics and Automation Letters (RA-L)*, 2018.

[78] B.-N. Vo, B.-T. Vo, and M. Beard, Multi-Sensor Multi-Object Tracking With the Generalized Labeled Multi-Bernoulli Filter, *IEEE Transactions on Signal Processing*, vol. 67, no. 23, 2019, pages: 5952–5967, 2019.

[79] N. Wojke, A. Bewley, and D. Paulus, Simple online and realtime tracking with a deep association metric, in *Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pages: 3645–3649.

[80] Y. Xia, K. Granstrcom, L. Svensson, and A. F. Garcia-Fernandez, Performance evaluation of multi-bernoulli conjugate priors for multi-target filtering, in *Proceedings of the 2017 20th International Conference on Information Fusion (Fusion)*, 2017, pages: 1–8.

[81] L. Xu, J. Jia, and Y. Matsushita, Motion Detail Preserving Optical Flow Estimation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, 2012, pages: 1744–1757, 2012.

[82] X. Xuezhi, M. Zhai, R. Zhang, Y. Qiao, and A. El Saddik, Deep Optical Flow Supervised Learning With Prior Assumptions, *IEEE Access*, vol. PP, Aug. 2018, pages: 1–1, Aug. 2018.

[83] C. Ye, A. Mitrokhin, C. M. Parameshwara, C. Fermüller, J. Yorke, and Y. Aloimonos, *Unsupervised Learning of Dense Optical Flow and Depth from Sparse Event Data*, Sep. 2018.

[84] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uricár, S. Milz, M. Simon, K. Amende, *et al.*, Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving, in *Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pages: 9308–9318.

[85] J. J. Yu, A. W. Harley, and K. G. Derpanis, Back to Basics: Unsupervised Learning of Optical Flow via Brightness Constancy and Motion Smoothness, in *Proceedings of the Computer Vision - ECCV 2016 Workshops, Part 3*, 2016.

[86] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, A survey of modern deep learning based object detection models, *Digital Signal Processing*, vol. 126, 2022, pages: 103514, 2022. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S1051200422001312`.

[87] A. Zanardi, A. Aumiller, J. Zilly, A. Censi, and E. Frazzoli, Cross-modal learning filters for RGB-neuromorphic wormhole learning, *Robotics: Science and System XV*, 2019, pages: P45, 2019.

[88] M. Zhai, X. Xuezhi, R. Zhang, L. Ning, and A. El Saddik, Optical Flow Estimation Using Channel Attention Mechanism and Dilated Convolutional Neural Networks, *Neurocomputing*, vol. 368, Aug. 2019, Aug. 2019.

[89] S. Zhao, X. Li, and O. E. F. Bourahla, Deep Optical Flow Estimation Via Multi-Scale Correspondence Structure Learning, Aug. 2017, pages: 3490–3496.

[90] Y. Zhou, G. Gallego, H. Rebecq, L. Kneip, H. Li, and D. Scaramuzza, Semi-dense 3D reconstruction with a stereo event camera, in *Proceedings of the Proceedings of the European conference on computer vision (ECCV)*, 2018, pages: 235–251.

[91] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, The Multivehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception, *IEEE Robotics and Automation Letters*, vol. 3, no. 3, 2018, pages: 2032–2039, 2018.

[92] A. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras, in *Proceedings of the Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, Jun. 2018.

[93] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, Unsupervised event-based learning of optical flow, depth, and egomotion, in *Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pages: 989–997.

COLLEGE SCIENCES
DOCTORAL DE L'INGENIERIE
PAYS DE LA LOIRE ET DES SYSTEMES

CENTRALE NANTES

Titre : Détection et suivi d'objets en mouvement à l'aide d'une vision hybride basée sur les événements et sur les images pour la conduite autonome

Mots clés : Caméra basée sur les événements, flux optique, détection d'objets en mouvement, suivi d'objets

Résumé : La caméra basée sur les événements est un capteur bioinspiré qui diffère des caméras à images conventionnelles : Au lieu de saisir des images à une fréquence fixe, elles surveillent de manière asynchrone les changements de luminosité par pixel et produisent un flux de données d'événements contenant l'heure, le lieu et le signe des changements de luminosité. Les caméras événementielles offrent des propriétés intéressantes par rapport aux caméras traditionnelles : haute résolution temporelle, gamme dynamique élevée et faible consommation d'énergie. Par conséquent, les caméras événementielles ont un énorme potentiel pour la vision par ordinateur dans des scénarios difficiles pour les caméras traditionnelles, tels que le mouvement rapide et la gamme dynamique élevée. Cette thèse a étudié la détection et le suivi d'objets avec la caméra événementielle en se basant sur un modèle et sur l'apprentissage profond. La stratégie de fuison avec la caméra d'image est proposée puisque la caméra d'image est également nécessaire pour fournir des informations sur l'apparence. Les algorithmes de perception proposés comprennent le flux optique, la détection d'objets et la segmentation du mouvement. Des tests et des analyses ont été effectués pour prouver la faisabilité et la fiabilité des algorithmes de perception proposés.

Title : Moving Objects Detection and Tracking using Hybrid Event-based and Frame-based Vision for Autonomous Driving

Keywords : Event-based camera, optical flow, moving object detection, object tracking

Abstract : The event-based camera is a bio-inspired sensor that differs from conventional frame cameras: Instead of grabbing frame images at a fixed rate, they asynchronously monitor per-pixel brightness change and output a stream of events data that contains the time, location and sign of the brightness changes. Event cameras offer attractive properties compared to traditional cameras: high temporal resolution, high dynamic range, and low power consumption. Therefore, event cameras have an enormous potential for computer vision in challenging scenarios for traditional frame cameras, such as fast motion, and high dynamic range. This thesis investigated the model-based and deep-learning-based for object detection and tracking with the event camera. The fusion strategy with the frame camera is proposed since the frame camera is also needed to provides appearance infomation. The proposed perception algorithms include optical flow, object detection and motion segmentation. Tests and analyses have been conducted to prove the feasibility and reliability of the proposed perception algorithms.