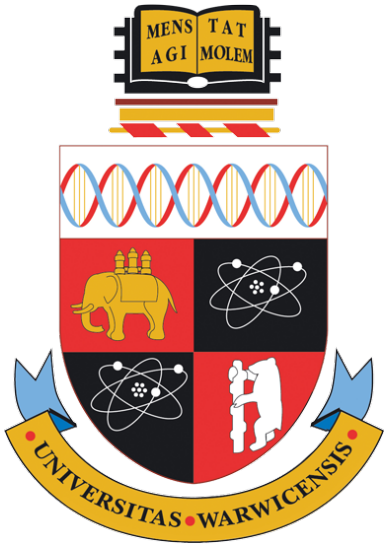
WARWICK
THE UNIVERSITY OF WARWICK



Gait Recognition with Event Cameras

by

Bowen Du

Thesis

Submitted to the University of Warwick

in partial fulfilment of the requirements

for admission to the degree of

Doctor of Philosophy

Department of Computer Science

Oct 2021

Contents

List of Tables

List of Figures

Acknowledgements

Firstly, I would like to express my sincerest gratitude to my supervisor Dr. Hongkai Wen, whose extensive knowledge and deep insights have helped and encouraged me during my research life at the University of Warwick. I have learnt a lot about how to be a good researcher and scientist under his guidance. I will keep moving forward, just like him, and look forward collaboration in the future.

I would like to thank Prof. Yiran Shen, Dr. Hongfei Fan, Prof. Miaomiao Zhang and Dr. Hongming Zhu. During my PhD period, they provided me with a lot of valuable research advice and help. Their help is not limited to academics, but also includes career and life-long development.

I thank my lab-mates and friends, Man Luo, Lichuan Xiang, Qingzhi Ma, Junyu Li, Haoyi Wang, Yijun Quan, Yujue Zhou, Wentai Wu, Jiashu Liao, Shuang Wang, Hao Wu, and Zhiyan Chen. The moments of discussing problems, sharing delicious food, playing basketball and getting together are also the important part of my PhD life.

Last but not least, I want to thank my parents sincerely. With their support and understanding, I can concentrate on my PhD journey without worrying about other issues. With them, I have the power to fight against difficulty and realise my dreams.

Declarations

The work in this thesis was developed and conducted by the author between October 2017 and June 2021. The author declares that, apart from work whose authors are explicitly acknowledged, this thesis and the materials contained in this thesis represent original work undertaken solely by the author. The author confirms that this thesis has not been submitted for a degree at another university.

1 Publications

Parts of this thesis have been previously published by the author in the following:

[62] Bowen Du, Weiqi Li, Zeju Wang, Manxin Xu, Tianchen Gao, Jiajie Li, and Hongkai Wen. Event encryption for neuromorphic vision sensors: Framework, algorithm, and evaluation. *Sensors*, 21(13):4320, 2021

[227] Yanxiang Wang, Bowen Du, Yiran Shen, Kai Wu, Guangrong Zhao, Jianguo Sun, and Hongkai Wen. Ev-gait: Event-based robust gait recognition using dynamic vision sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6358–6367, 2019

[228] Yanxiang Wang, Xian Zhang, Yiran Shen, Bowen Du, Guangrong Zhao, Cui Lizhen, and Hongkai Wen. Event-stream representation for human gaits identification using deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021

Research was performed in collaboration during the development of this thesis, but does not form part of the thesis:

[61] Bowen Du, Chris Xiaoxuan Lu, Xuan Kan, Kai Wu, Man Luo, Jianfeng Hou, Kai Li, Salil Kanhere, Yiran Shen, and Hongkai Wen. Hydradoctor: real-time

liquids intake monitoring by collaborative sensing. In *Proceedings of the 20th International Conference on Distributed Computing and Networking*, pages 213–217, 2019

[191] Yiran Shen, Bowen Du, Weitao Xu, Chengwen Luo, Bo Wei, Lizhen Cui, and Hongkai Wen. Securing cyber-physical social interactions on wrist-worn devices. *ACM Transactions on Sensor Networks (TOSN)*, 16(2):1–22, 2020

[137] Chris Xiaoxuan Lu, Bowen Du, Xuan Kan, Hongkai Wen, Andrew Markham, and Niki Trigoni. Verinet: User verification on smartwatches via behavior biometrics. In *Proceedings of the First ACM Workshop on Mobile Crowdsensing Systems and Applications*, pages 68–73, 2017

[138] Chris Xiaoxuan Lu, Bowen Du, Hongkai Wen, Sen Wang, Andrew Markham, Ivan Martinovic, Yiran Shen, and Niki Trigoni. Snoopy: Sniffing your smartwatch passwords via deep sequence learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):1–29, 2018

[139] Chris Xiaoxuan Lu, Bowen Du, Peijun Zhao, Hongkai Wen, Yiran Shen, Andrew Markham, and Niki Trigoni. Deepauth: in-situ authentication for smartwatches via deeply learned behavioural biometrics. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, pages 204–207, 2018

[233] Hongkai Wen, Ronald Clark, Sen Wang, Xiaoxuan Lu, Bowen Du, Wen Hu, and Niki Trigoni. Efficient indoor positioning with visual experiences via lifelong learning. *IEEE Transactions on Mobile Computing*, 18(4):814–829, 2018

[190] Yiran Shen, Fengyuan Yang, Bowen Du, Weitao Xu, Chengwen Luo, and Hongkai Wen. Shake-n-shack: Enabling secure data exchange between smart wearables via handshakes. In *2018 IEEE international conference on pervasive computing and communications (PerCom)*, pages 1–10. IEEE, 2018

[140] Chris Xiaoxuan Lu, Peijun Zhao, Bowen Du, Hongkai Wen, Andrew Markham, Stefano Rosa, and Niki Trigoni. Automatic face recognition adaptation via ambient wireless identifiers. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 377–378, 2018

[141] Chris Xiaoxuan Lu, Xuan Kan, Bowen Du, Changhao Chen, Hongkai Wen, Andrew Markham, Niki Trigoni, and John Stankovic. Autonomous learning for face recognition in the wild via ambient wireless cues. In *The World Wide Web Conference*, pages 1175–1186, 2019

2 Sponsorships and Grants

This research was funded by the Chancellor's International Scholarship.

Abstract

Gait recognition is a fundamental task in activity tracking, health monitoring, security surveillance and many other computer vision applications. A variety of sensors have been utilised for gait recognition, such as standard cameras, infrared cameras, floor sensors and inertial sensors. However, each kind of sensor has its limitation by nature. Event camera is a new bio-inspired vision sensor with lower energy consumption, broad dynamic range, and high temporal resolution. These advantages enable event cameras to be suitable for surveillance tasks, especially under special conditions such as long-term, sensitive, and challenging lighting scenarios. Unfortunately, to the best of our knowledge, there has been no event-based gait recognition technique available before. In this thesis, we focus on enabling approaches and solutions on *gait recognition with event cameras*. Firstly, due to the lack of relevant data, we produce two event-based gait datasets using an event camera, which serve as a basis for model training as well as quantitative evaluations and comparisons. Secondly, we propose a CNN-based approach named *EV-Gait* which achieves gait recognition with event cameras, and devise a scheme that includes image-like representation, noise cancellation and a neural network. Thirdly, we further propose a GCN-based *3DGraph-Gait* approach that extracts spatiotemporal features from event streams, which improves the accuracy of recognition, and enables real-time gait recognition that only requires a limited number of events generated in several milliseconds. Finally, since privacy is a major concern with gait recognition, we propose an encryption framework named *EV-Encryp*, which effectively protects personal privacy and meanwhile, preserves the efficiency of the follow-up gait recognition after decryption. In summary, this study has initialised a novel research direction namely gait recognition with event cameras, contributed innovative supporting techniques and solutions, and established key foundations for further exploration and extensions.

Acronyms

APS Active Pixel Sensor.

ATIS Asynchronous Time Based Image Sensor.

CMOS Complementary Metal-Oxide-Semiconductor Transistor.

CNN Convolutional Neural Network.

DAVIS Dynamic and Active-Pixel Vision Sensor.

DVS Dynamic Vision Sensor.

GAN Generative Adversarial Network.

GCN Graph Convolutional Network.

GMM Gaussian Mixture Model.

IMU Inertial Measurement Unit.

LSTM Long Short-Term Memory.

NVS Neuromorphic Vision Sensor.

RNN Recurrent Neural Network.

SLAM Simultaneous Localisation and Mapping.

SVM Support Vector Machine.

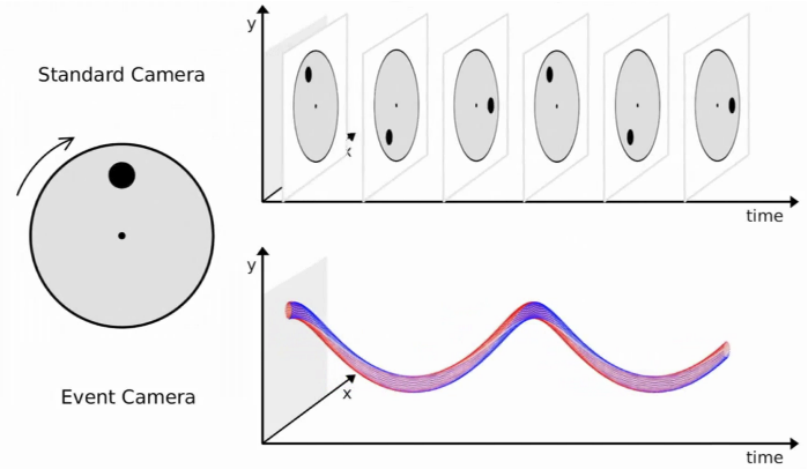
Chapter 1

Introduction

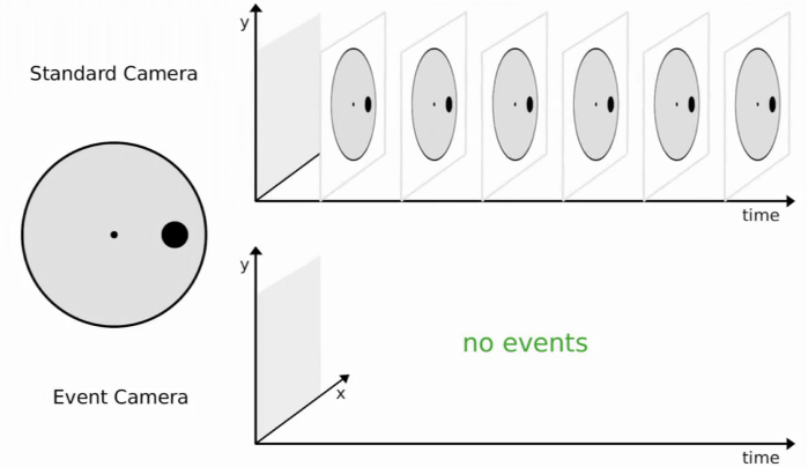
1.1 Background

In the recent decade, the technology of computer vision has been developing rapidly, and many achievements have been made in this area, such as object recognition [136, 215], human identification [54, 147, 158, 220], super-resolution image generation [11, 230], and simultaneous localisation and mapping (SLAM) [184, 208]. Benefiting from the growing computational capability and abundant data, deep learning has further broadened the boundary of computer vision. However, some inherent limitations of traditional cameras have affected their applications in some specific scenarios. For example, images will be blurry in a high-speed movement, and the quality of images will decrease seriously in dark or extremely bright lighting conditions. Some novel vision sensors, including event cameras, modulo cameras and infrared cameras, have emerged with the development of computer vision to overcome such drawbacks.

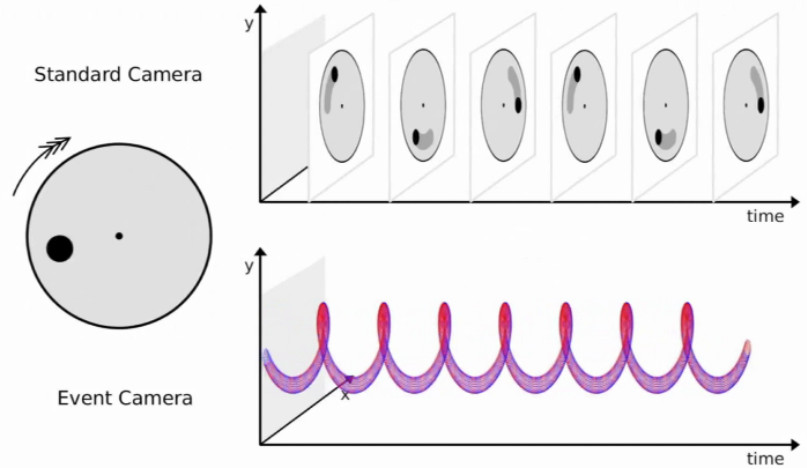
The event camera, also known as the neuromorphic vision sensor (NVS), was initially proposed in 1991 [143]. It imitates the impulse generation mechanism of neurons on the retina, making it possible to sense the visual changes more sensitively, and thus, it is also called the silicon retina. Compared with a traditional camera, an event camera responds to the illumination intensity variation of each pixel independently, rather than producing a full image at a fixed rate. Their comparisons are visualised in Figure 1.1. Under regular conditions (see Figure 1.1 (a)), the standard camera outputs full images with intensities in every pixel, while the event camera produces a stream



(a) Regular moving



(b) Static scenario



(c) High-speed moving

Figure 1.1: Output comparison between standard cameras and event cameras[179]

of events that represent the intensity changes in each pixel. In a static scenario when there is no movement, the event camera does not have any output, but the standard camera still generates images at a fixed rate, which is shown in Figure 1.1(b). When an object is moving at a high speed, the standard camera will inevitably produce blurry images due to the synchronous exposure, but the event camera will generate the standard and correct event stream as usual, as shown in Figure 1.1(c).

This specific imaging mechanism brings event cameras many advantages. Each pixel of an event camera can capture the change of illumination intensity asynchronously in several tens of microseconds, which significantly surpasses the temporal resolution of traditional cameras. In addition, event cameras also have a broader dynamic range and lower energy consumption, bringing computer vision into new scenarios that traditional cameras cannot or are difficult to touch. For example, drones and robots capture surrounding information and conduct localisation in unstable, high-speed moving conditions, and vehicles sense neighbour vehicles, obstacles and pedestrians when entering or leaving tunnels. However, there is still a long journey to fully employ the advantages of event cameras because of the lack of specialised techniques. The approaches and algorithms for traditional frame-based images cannot be directly used for event cameras, because the output format of event cameras, a stream of events that encode the intensity changes of each pixel, is totally different from the conventional image-like data. Designing dedicated solutions for event cameras can make the most use of their specific characteristics, and achieve better performance on traditional computer vision tasks.

Gait recognition is a fundamental task in a lot of real-world applications, such as security surveillance, activity tracking and health monitoring. Similar to other biometric identification approaches such as face recognition, iris recognition and fingerprint recognition, gait recognition identifies an individual by his/her walking patterns. Because it can work remotely without closely touching targets, it can be applied in many new scenarios and solve novel challenges. For example, in an environment with a potential COVID-19 threat, gait-based identification can complete the task while preventing the spread of the virus. Although there exist some gait recognition approaches based on RGB cameras, floor sensors, wearable sensors and

radar, these sensors have their limitations for gait recognition. For example, RGB cameras cannot work well in challenging light conditions, and wearable inertial sensors should be equipped for everyone who needs to be recognised.

Energy consumption, challenging light condition and motion blur are still challenges for gait recognition using traditional cameras. Firstly, in most scenarios, gait recognition is commonly associated with round-the-clock (24hrs) surveillance, where the energy consumption is an important issue to be considered, and lower energy consumption will benefit the applications of gait recognition. Secondly, due to the fundamental imaging mechanism of traditional cameras, gait recognition algorithms cannot work well under strong light or dark conditions based on such cameras. Thirdly, an individual's high-speed walking leads to a motion blur, which hampers the accuracy of gait recognition.

It is fortunate that the natural advantages of event cameras bring the possibility to over-come these problems. Its unique working mechanism allows event cameras only to re-spond to the change of illuminate intensity, which saves much energy compared with generating a full-size image every several milliseconds. Furthermore, some features of event cameras, i.e., broad dynamic range and high temporal resolution, will mitigate the effects of poor light conditions and motion blur for gait recognition. However, as an emerging sensor, event cameras have not been used for gait recognition, and the main challenges for applying event cameras in gait recognition are the lack of available datasets and techniques.

In order to complete the gait recognition task using event cameras, the highest priority is to collect enough event-based gait data. On the one hand, these data can help researchers understand event-based gait patterns, and further serve as a benchmark for quantitative evaluations and comparisons. On the other hand, models can be trained for gait recognition directly using these data. Just like the benefits brought by the ImageNet dataset, event-based gait data will boost the development of gait recognition and applications of event cameras.

With enough gait images, neural networks have been proven as an effective approach for gait recognition using standard RGB cameras. These neural networks for gait recognition can extract relevant features from images or videos and perform

classification. However, the outputs of event cameras do not resemble images, and thus existing neural networks for traditional cameras cannot be used for event cameras directly. A possible solution is converting the outputs of event cameras into an image-like format, which can reuse the exiting image-based recognition algorithms. Another approach is to design a dedicated representation and the corresponding neural network. This representation should express the gait-related patterns of event cameras' outputs, and the neural networks need to extract these patterns. In addition, the high temporal resolution is one of the advantages of event cameras, enabling real-time gait recognition that only requires a limited number of events in several milliseconds.

Security and privacy are major concerns with gait recognition, but these problems for event cameras have not been explored before. Gait, as a biometric characteristic, also deserves protection against unauthorised access. Meanwhile, some information that is irrelevant to gait recognition is also captured during surveillance, which may put personal privacy at a risk. Prior work supposed that it is secured because event cameras generate streaming data rather than visual images. However, some reconstruction algorithms, which generate visual images from an event stream, may threaten the privacy related to the applications of event cameras, including gait recognition. Furthermore, the event streams should also be securely transmitted and stored for security requirements. Although there are some encryption schemes for traditional images, videos and other visual data, these schemes are not suitable for event data. On the one hand, the formats of image or video are totally different from that of the event stream, and traditional encryption schemes are not suitable for event cameras. On the other hand, event cameras may generate more than millions of events per second, and the efficiency of the encryption scheme is another major concern, which affects the data transmission in real-time surveillance.

1.2 Thesis Contributions

This thesis focuses on enabling approaches and solutions to gait recognition with event cameras. Since there is no gait dataset captured by event cameras, the first step to solve this problem is to collect event-based gait data, which serve as a basis

for model training as well as quantitative evaluations and comparisons. After preparing adequate data, dedicated approaches are devised for gait recognition. For representation methods, event streams are expressed in image-like or point-cloud-like formats, respectively. For the architecture of models, some neural networks that have achieved much success for images and point clouds are utilised for event streams to extract features. Furthermore, the ensemble network, which makes use of the high temporal resolution of event cameras, can perform recognition using events in several milliseconds while maintaining competing accuracy with the entire event streams. In addition to effectiveness and efficiency, the security and privacy issues also are well resolved. The main contributions of this thesis can be summarised as follows:

(i) In observation of the lack of available datasets, we produce two event-based gait datasets. The first dataset is called DVS128-Gait, which is captured by the event camera DVS128 in real-world settings. Another dataset is EV-CASIA-B, which is generated from the widely used RBG benchmark, CASIA-B. Further quantitative evaluations are conducted based on these datasets, and relevant models are also trained using the data. These datasets allow event cameras to be applied in gait recognition.

(ii) With enough event-based gait data, a novel event-based gait recognition approach, EV-Gait, is proposed, which is specifically designed for event cameras. It is able to effectively remove noise in event streams by enforcing motion consistency and employs a CNN to recognise gait from the asynchronous and sparse event data. This approach explores a possible way to bring existing state-of-the-art algorithms for image/video-based gait recognition into the domain of event cameras.

(iii) Making the most use of the event stream's spatiotemporal feature, a 3D-graph-based gait recognition approach, 3DGraph-Gait, is proposed. At the beginning, a graph-based representation method projects the event stream into three-dimensional spatiotemporal space and constructs a graph according to the distance between events. Then a GCN-based model is trained after converting the event streams to graphs. Finally, this model is further extended to recognise the

limited number of events, which fully uses the high temporal resolution of event cameras.

(iv) To protect the application of event cameras in gait recognition and other private scenarios, EV-Encryp, an encryption framework for event cameras, is devised. Firstly, a general threat model for event cameras is proposed, which defines the adversary's objectives, capabilities and knowledge, and the encryption framework can prevent attacks included in this threat model. The encryption framework utilises the chaotic maps to generate pseudo-random sequences to shuffle the position and polarity information. Because event cameras can generate more than millions of events per second, the framework balances the expense of encryption and the risk of security, and dynamically controls the encryption process according to the type of event cameras and the devices.

1.3 Thesis Outline

The rest of this thesis is organised as follows:

In Chapter 2, related work is reviewed, including event cameras, gait recognition approaches and visual data encryption schemes. Firstly, the mechanism, features and applications of the event cameras are summarised, which provide an overview of the capabilities of event cameras. Then, the development of gait recognition is reviewed and analysed, which involves various approaches, sensors, and scenarios. Finally, some encryption schemes for vision sensors are summarised and compared.

In Chapter 3, a novel event-based gait recognition approach, EV-Gait, and two event-based gait recognition datasets are proposed. Firstly, the image-like representation method is described, which is inspired by the mechanism of the event camera. Secondly, considering the effects of numerous noises in event streams, a velocity-based noise cancellation method is designed to improve the data quality of the event stream. Thirdly, a CNN-based approach is utilised to extract gait-related features from the image-like representation and recognise identities. Finally, the synthesised gait dataset, EV-CASIA-B, and the real-world dataset, DVS128-Gait, are presented and utilised for quantitative evaluation.

In Chapter 4, a GCN-based gait recognition approach, 3DGraph-Gait, is devised, which can extract spatiotemporal features for event streams and recognise identities using a limited number of events only. The graph-based representation of the event stream and sampling strategy are the fundamental components of 3DGraph-Gait. After sampling and representing, the GCN-based model is trained using the processed event streams, and an ensemble network is further devised to improve the real-time performance using a small partition of the streams. At last, a comprehensive set of experiments are conducted to evaluate (i) the impacts of 3DGraph-Gait's hyperparameters on the performance using the entire event streams, (ii) the performance of 3DGraph-Gait compared with other approaches and (iii) the performance of 3DGraph-Gait only using a limited number of events.

In Chapter 5, an event-oriented efficient encryption framework, EV-Encryp, is proposed to secure event-based applications, such as gait recognition and other privacy-related scenarios. The pseudo-random sequence is generated by chaotic maps and is employed to scramble events' position and polarity information. An indicator, updating score, is designed to balance the effectiveness and security for different devices and event cameras with different resolutions. A set of experiments are conducted to evaluate the effectiveness and efficiency of the proposed encryption framework.

In Chapter 6, we summarise the major achievements for gait recognition with event cameras in this thesis, identify more issues that could be further resolved and explore the applications of event cameras in the future.

Chapter 2

Literature Review

2.1 Event Cameras

2.1.1 Introduction to Event Cameras

The prototype of event cameras, the silicon retina [143], was initially proposed in 1991. This new bio-inspired chip was designed according to the neural architecture of the human's eyes. With this chip, it is possible to explore a more powerful computing approach for the vision system, and several asynchronous event-driven imagers occurred [106, 123–125], whose resolutions increase from 32×32 to 128×128. Until 2008, the first generation off-the-shelf event camera, DVS128 [126], was available, promoting the development of event camera-based applications. From then on, there came out more and more commercial event cameras by different companies. For example, Prohesee's ATIS [172], Samsung's VGA-DVS [197] and CelePixel's CeleX [44]. Now, there are several event cameras whose resolutions can reach one million [44, 66], approaching the resolution of traditional cameras. In addition to the rapid increase in resolution, other visual/inertial modalities are also fused into event cameras [26, 213], extending their applications in real-world scenarios.

Compared with traditional RGB cameras, which produce synchronised frames at fixed rates, the pixels of event cameras can capture microseconds level intensity changes independently and generate a stream of asynchronous 'events'. These events resemble the impulses in the human's nervous system, and thus, the event camera

is also called the neuromorphic vision sensor (NVS). Similar to the generation of impulses, an event is produced when the logarithmic intensity change of one pixel reaches the predefined threshold. At the same time, the corresponding pixel records its logarithmic intensity and keep monitoring the change of itself [71]. An event can be described as a quadruplet, (t, x, y, p), where t is the timestamp when the event happens, (x, y) is the location of the event in the 2D pixel space, and $p \in \{+1, -1\}$ is the polarity of the event. The generation of an event can be formulated as:

$$p = \begin{cases} +1, \text{ if } \log(I_{x,y,t}) - \log(I_{x,y,t-\Delta t}) = C^{+} \\ -1, \text{ if } \log(I_{x,y,t}) - \log(I_{x,y,t-\Delta t}) = -C^{-} \end{cases} \tag{2.1}$$

where $I_{x,y,t}$ is the current intensity of a pixel located at (x, y), the t is the time when the current event happens, Δt is the time duration from the last event happened at the same pixel, C^{+} and C^{-} are the predefined positive and negative thresholds to determine the positive and negative events respectively.

Benefiting from the generation mechanism of events, event cameras outperform traditional RGB cameras in several aspects. Firstly, event cameras require much less resources including energy, bandwidth and computation as the events are sparse and only triggered when intensity changes are detected. For example, the DVS128 sensor consumes 150 times less energy than a Complementary Metal-Oxide-Semiconductor Transistor (CMOS) camera [126]. Secondly, the temporal resolution of event cameras is tens of microseconds which means the they are able to capture detailed motion phases or high speed movements without blur or rolling shutter problems. Finally, event cameras have significantly larger dynamic range (up to 140dB [126]) than RGB cameras (about 60dB), which allows them to work under more challenging lighting conditions. These characteristics make event cameras more appealing over RGB cameras for vision tasks with special requirements on latency, resources consumption and operation environments.

Although event cameras hold many advantages compared with standard cameras, there are still some problems and challenges for event cameras to be utilised in more general scenarios. First of all, the generated event streams are very noisy. In practice, those cameras are very sensitive to illumination changes or perturbation in

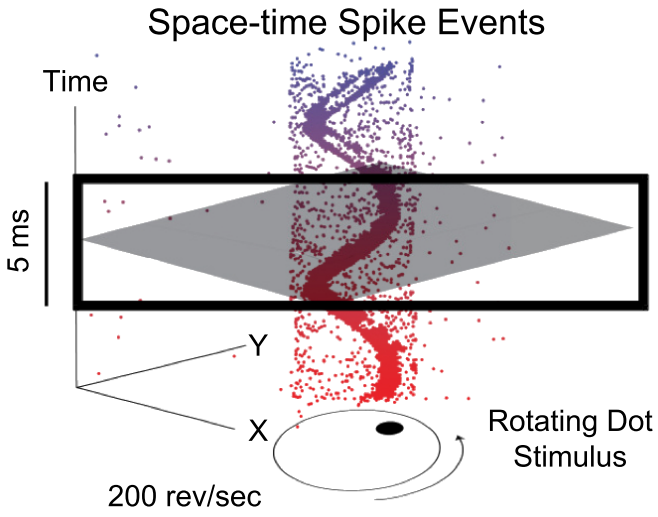


Figure 2.1: An event stream caused by a rotating dot (adapted from [131]).

the background, and often report a large number of events that are not relevant to the moving objects. For example, as can be seen in Figure 2.1, although there is only a rotating dot in the scene, the resulting event stream contains many ad-hoc events that are detached from the desired spiral. This tends to have a significant negative impact on the performance of various applications, which hinders the wide adoption of event cameras. Secondly, effective and efficient approaches to process event streams have not been fully explored. A single event only contains binary intensity change information and timestamp. Hence, how to aggregate visual information and make the most of temporal information is a key problem to be solved for event cameras. Thirdly, some dedicated algorithms/neural networks with event cameras should be designed. The CNN-based neural network can be a backbone for different kinds of tasks for traditional cameras, but there is no recognised backbone and algorithm for event stream. Finally, the approach of how event cameras work together with other kinds of sensors has not been fully investigated. Event cameras provide a new modality of visual information, and some complex tasks utilise several different sensors. Data fusion approaches for event cameras and other sensors benefit these complex tasks.

2.1.2 The Applications of Event Cameras

With the development of the event camera, it has been managed to solve different kinds of computer vision tasks. Feature detection and tracking is the basic component for

various high-level vision tasks, and event cameras also have some achievements in this area. For corner detection, the space-time properties of moving edges are employed to estimate planes, and further, the corners can be extracted from the planes [47]. Learning-based approaches, such as random forest [145], have been employed to recognise corner events from the event stream. Inspired by the corner detection algorithms for standard cameras, including Harris [79] and FAST [181], event-based Harris [216] and FAST [153] are designed, respectively. For tracking the event-based features, a graph is utilised to track the movement of the accurately detected corners [9]. Multikernial-based algorithm [110] and particle filter [73] are also used for this task. Based on these achievements for event cameras, some high-level computer vision tasks can be explored accordingly.

Recognition is another important task for computer vision, which involves feature extraction from the raw data and feature-based comparison/classification. HOTS is the hierarchy of event-based time surface, which can be used to describe a short stream of events. After presenting the events as HOTS, a classifier can be trained to recognise letters, digits, and even human faces [111]. HFirst combines Gabor filers, templated matching and classifier to build a neural network to solve the recognition problem [163]. Histograms of Averaged Time Surfaces (HATS) has been proposed as a new description for recognition, which still uses temporal information rather than polarity-related information [195]. Inspired by the attention mechanism [218] applied in natural language processing, this component is also employed to improve the performance of event-based object recognition [36]. In addition to these relatively simple recognition tasks, some event-based recognition algorithms are proposed for real-world applications, such as pedestrian detection [41, 94, 161], gesture recognition [43, 114] and activity recognition [38, 89].

Since there are various algorithms and approaches for standard cameras, a straightforward way to bring these algorithms to the area of event cameras is to reconstruct images or videos from event streams. Taking the intensity change information of events, the approach that managed to estimate the intensity mosaic firstly uses pixel-wise incremental Extended Kalman Filter (EKF) to compute the log gradient for each pixel, and then combine Poisson reconstruction to generate the absolution log

intensity [100]. The image construction can also be treated as a visual interpretation problem, which employs a network to figure out a mutually consistent state, and the state can provide the visual interpretation, including intensity image [50]. In [17], a patch-based dictionary is learnt from event streams offline, and reconstruction is executed online based on the dictionary. A variational model is proposed to estimate the behaviour of event cameras, and the grayscale images are reconstructed from the model [155]. A self-supervised learning approach is proposed in [167], which combines optical flow and event-based photometric constancy.

In addition to grayscale image reconstruction, some video synthesis algorithms have been further proposed. Events-to-video, E2VID [177], is an end-to-end neural network-based approach, which is trained with the data generated from the simulator. This approach shows a good generalization with real-world data. Generative Adversarial Networks (GANs) have been used to generate videos from event streams. Both conditional GAN [223] and enhanced Cycle-GAN [242] have shown their capabilities in generating high-quality videos. Furthermore, some work has successfully generated super-resolution intensity images from events. EventSR [224] utilises three neural networks to complete reconstruction, restoration and super-resolution tasks, respectively. Besides, another end-to-end neural network for super-resolution reconstruction is proposed in [46], which pairs the events and the optical flow to generate images.

Apart from the aforementioned applications, event cameras have also been applied in optical flow estimation [16, 24, 70, 168, 182, 255], 3D reconstruction [23, 103, 104, 171, 186, 256], SLAM [37, 101, 108, 151, 176, 232] and motion segmentation [148, 201, 217]. Event cameras are also equipped on robots [49, 56, 57, 152] and drones [58, 200] to perceive the surrounding environment under more challenging conditions. This new vision sensor is expected to solve some problems that traditional cameras cannot complete.

2.1.3 The Datasets of Event Cameras

As numerous datasets of standard cameras improve the development of computer vision, the increasing number of algorithms and applications of event cameras benefit

from the high-quality event-based datasets. These datasets are captured for general or specific tasks and leveraged for training and evaluation, which include:

- **Neuromorphic-MNIST (N-MNIST) Dataset [162].** The MNIST dataset is a fundamental dataset to investigate traditional computer vision tasks and evaluate their performance. N-MNIST dataset uses the regular moving ATIS sensor to capture the MNIST images displayed on an LCD monitor. It consists of 60,000 training and 10,000 testing samples with only raw events. As the earliest event camera dataset, some object detection [35] and classification tasks [27, 195] are evaluated using this dataset.

- **DAVIS 240C Dataset [154].** This dataset is captured using a DAVIS240C camera, which contains the raw events, grayscale images, IMU measurements, and camera calibration from the DAVIS and the pose-oriented ground truth from a motion-capture system. This dataset involves 25 sequences, whose scenarios include laboratory, office, campus and urban areas. As a general event camera dataset, DAVIS240C dataset has been applied in various types of tasks, such as visual odometry [69, 219] and image reconstruction [177].

- **DAVIS Driving Dataset 2017 (DDD17) [29].** DDD17 is an automotive driving dataset using a DAVIS camera. Besides the events, frames and IMU measurements from the DVAIS camera, other driving-related data, such as steering angle, engine speed, fuel level, the state of the parking brake and GPS etc., are also recorded. This dataset covers various driving areas, such as downtown, freeway and campus, under different lighting and weather conditions, including day, evening, night, rain, sunny and wet. There are about 36 sequences included in this dataset, and over 12 hours of data are recorded. This larger dataset provides an opportunity to design an end-to-end neural network for intelligent driving. An event-based steering prediction neural network is trained based on this dataset and can achieve better prediction than using traditional grayscale images [146]. In addition, this dataset has been used for semantic segmentation [7, 225]. An extension of DDD17, DDD20 [87], is published recently, including 51 hours of driving data.

- **Multi Vehicle Stereo Event Camera (MVSEC) Dataset [257].** MVSEC dataset is collected on four different vehicle platforms, including hexacopter, handheld, car and motorcycle, under different environment (indoor and outdoor) and lighting (day and night) conditions. Two DAVIS 346B cameras are equipped on the left and right sides of the vehicle to capture events, grayscale images and IMU measurements as the modality generated from the event camera. In addition, a visual-inertial sensor (including left and right cameras, one IMU), a VLP-16 lidar and a GPU module are equipped. Furthermore, a motion capture system is utilised to provide the ground truth of the movement. The multi-modality data involved in this dataset allows the researcher to develop cross-modality fusion algorithms or compare the task performance using different modalities. Some optical flow estimation models [113, 166, 255] are trained or evaluated based on this dataset.

- **DVS Human Pose Estimation (DHP19) Dataset [34].** DHP19 is the first human pose dataset captured by four synchronised DVS cameras. Although the DAVIS346 event camera is utilised for this dataset, only DVS outputs (i.e., events) are reserved due to the host's USB bandwidth limitation. The Vicon motion capture system provides the ground truths of the captured poses. There are a total of 33 movements from 17 volunteers (12 female and 5 male). These 33 movements, such as left/right arm/leg abduction, jumping, punching, kicking etc., are classified into five different classes. This dataset enables a more detailed human action analysis using an event camera. This dataset has been employed to evaluate the performance of 2D pose estimation [258] and gesture classification [43].

- **DVS Noise (DVSNOISE20) Dataset [15].** DVSNOISE20 is released with an event-based denoising algorithm and collected for quantitative evaluation on the performance of denoising. It is captured by the DAVIS346 event camera, whose movement is restricted by a gimbal, and additional IMU measurements are provided. The ground truth of noise events is computed according to the grayscale images from DAVIS and velocity from IMU. These labelled events

can be used to train a denoising neural network [15], and to evaluate various denoising algorithms as a benchmark [63].

Although several datasets of event cameras have been released, the number of available datasets which can be used to train and evaluate neural networks are still limited. Because of the widespread traditional cameras, many images and videos are generated every second in our daily lives. Compared with these traditional cameras, before event cameras will become commonly used sensors, more datasets are required to extend the application of event cameras, especially for the tasks that can demonstrate their unique advantages.

2.2 Gait Recognition

2.2.1 Introduction to Gait Recognition

Gait recognition is a kind of biometric authentication approach, which identifies a person by his/her manner of walking [220]. Compared with other biometric traits (e.g., face [82], iris [55] and fingerprints [144]), gait can be captured remotely with simple instrumentation using a low-resolution camera and is also difficult to be impersonated. These advantages make gait recognition an important biometric identification approach. The first gait recognition approach was proposed by Sourabh N. et al. in 1994 [159], which used spatiotemporal patterns to analyse and recognise the walking person. A leg movement-based gait recognition approach was designed in 1997, which focused on the inclination of legs and the corresponding harmonic motion [52]. The stride length and cadence patterns were also considered to identify an individual's walking because they are affected by person's height, weight, and gender. These patterns were employed for gait recognition in 2002 [22]. Then, the silhouette was extracted from the videos to investigate gait recognition, avoiding the negative effects of the background [221]. A static and dynamic fusion approach was proposed in 2004, which combined the silhouettes and the joint-angle trajectories of lower limbs as the features for gait recognition [222]. In addition to these carefully picked features and patterns, Gait Energy Image (GEI) [78] and Gait History Image (GHI) [130], two

important model-free approaches, which did not directly model the structure of the human body, were proposed in 2006 and 2007, respectively. Recently, more and more deep learning-based approaches [8, 192, 234, 236] have been proposed to extract features and produce classification automatically.

The algorithms for gait recognition can be generally divided into two phases, feature extraction and comparison/classification. The target of feature extraction is to generate gait-related features/patterns from the raw data. These features include some static features, such as the skeleton [160], appearance silhouette [221], and various dynamic features, including the trajectory of joints [222] and stride length [22]. In addition, some latent/high-dimensional patterns can be learnt from neural networks [39, 67, 187, 188]. After extraction, the features are employed to identify the individual. Distance [33] and correlation [199] are firstly utilised to evaluate the similarity of each record. Then, machine learning-based classifier (e.g. SVM [142], decision tree [60] and neural networks [202]), hidden Markov model [97] and Bayesian classifier [21] are applied for classification. Combined with these approaches, until now, the accuracy of gait recognition can achieve about 90.4% accuracy [127] in CASIA-B [243], a famous gait recognition benchmark.

The development of gait recognition is always associated with high-quality datasets. On the one hand, the real-world gait datasets provide a benchmark for various approaches. On the other hand, researchers can investigate the robust algorithms or train neural networks under various scenarios and different settings (e.g., different views and clothes) using these datasets. The CMU Motion of Body (MoBo) dataset [75] was the first gait dataset published in 2001. After that, SOTON [193], CASIA-A [221], USF HumanID [185], CASIA-B [243] and CASIA-C [209] were released from 2002 to 2006, where CASIA-B is one of the most widely used datasets. CASIA-E [250] and OU-MVLP Pose [10] are available recently, showing that the gait recognition still attracts attention after several decades' development.

Standard gait recognition employs cameras to capture the movement, and vision-based approaches are designed to extract features. Nowadays, many other sensors have been applied for gait recognition, extending the application scenarios of gait recognition. Gait recognition can also be applied to wearable sensors. For example,

smartwatches and smart bands can identify the user by his/her gait. The accelerometer inside these wearable devices can detect the acceleration during the user's walking. Accelerometer-based recognition systems have been developed [59, 68, 180]. The vision-based techniques need monitoring a particular area, where floor sensors can also be equipped for gait recognition [91, 205, 206]. In addition, radars can capture the Doppler signatures of body components' movement, achieving high accuracy for identifying targets [164].

Gait recognition has potential values in real-world applications, and various approaches based on different sensors have managed to solve this problem. Nevertheless, the appearance of new sensors will bring new possible approaches to achieve better performance. These existing approaches can inspire the feature extraction and classification for event cameras and other new sensors.

2.2.2 Feature Extraction for Gait Recogntion

Feature extraction is a critical component for gait recognition, as the extracted features seriously affect the result of the later classification task. The feature extraction approaches can be roughly divided into two categories, model-based and model-free. Model-based approaches usually model the human body and extract features according to the model. In contrast, model-free approaches treat the whole motion or trajectory as integration and indirectly extract gait-related features. Although deep neural networks do not solely extract features, deep learning-based feature extraction blocks are treated as a new approach out of these categories because the extracted high-dimensional features are more complex than that of traditional model-free approaches.

For model-based approaches, the stride length and cadence, which represent steps per minute are firstly utilised to quantitatively describe an individual's walking patterns and recognise different identities [22]. Two-dimensional stick figures are used to describe the gait process, which is obtained by linking the nine body points (such as neck, shoulder, waist, pelvis, knees and ankles) [241]. Some distances between different components of the body, for example, left-right-foot, head-foot distances, are modelled for gait recognition [32]. The human movement can be abstracted as the movement of joints, and thus, the trajectories of joint angles are considered dynamic

information for recognition [210]. Furthermore, a fusion approach that combines the static and dynamic features achieves more accurate results [222]. Apart from the body features, some leg-based features are also extracted for gait recognition. Two new approaches, coupled oscillators and the biomechanics of human locomotion, are employed to model the thigh and leg inclination during walking and running and achieve higher accuracy in the running scenario [239]. A Fourier series is utilised to describe the motion of the upper leg, and temporal evidence gathering techniques are employed to extract the model from a sequence of images [53].

Model-free feature representation pays more attention to the global motion or trajectory, and the silhouette is a direct way to describe the movement of humans without the effects of background. The baseline algorithm proposed with HumanID treats the sequence of silhouettes as features for gait recognition [185]. In addition to directly using the silhouette, Motion-History Images (MHI) and Motion-Energy Images (MEI) are extracted from the sequence of silhouettes as the movement features [31]. If the binary image sequence that represents the areas of motion is $D(x, y, t)$, the MEI can be defined as

$$MEI_{\tau}(x, y, t) = \bigcup_{i=0}^{\tau-1} D(x, y, t - i) \tag{2.2}$$

where τ is the time duration to accumulate the motions. MEI is only a binary image, which cannot describe some temporal features. Based on the MEI, MHI is supplemented to such features and defined as:

$$MHI_{\tau}(x, y, t) = \begin{cases} \tau & if\ D(x, y, t) = 1 \\ max(0, MHI(x, y, t - 1) - 1) & otherwise. \end{cases} \tag{2.3}$$

Similarly, Gait Energy Images (GEI) [78] and Gait History Images (GHI) [130] are also constructed based on the silhouette sequence, but they use the means instead of the union. The GEI can be computed as:

$$GEI(x, y) = \frac{1}{N} \sum_{i=1}^{N} D(x, y, t) \tag{2.4}$$

GHI combines GEI and MHI, and is defined as:

$$GHI(x,y) = \begin{cases} \tau & if\ S(x,y) = 1 \\ \sum_{t=1}^{\tau} D(x,y,t) \cdot (t-1) & otherwise. \end{cases} \tag{2.5}$$

where $S(x,y)$ can be obtained from the intersection of $D(x,y,t)$ as follow:

$$S(x,y) = \bigcap_{t=1}^{\tau} D(x,y,t) \tag{2.6}$$

In addition, Frame Difference Energy Images (FDEI) [40] and Active Energy Images (AEI) [247] are also the two-dimension feature representations based on the silhouette sequence, focusing on the incomplete frames and the relations among gait cycle frames, respectively.

Whatever model-based or model-free approaches need to manually design proper features and extract them from the raw videos. Learning-based approaches provide an opportunity to generate features from the raw images automatically. The features extracted using convolutional neural networks also achieve high accuracy for view-invariant gait recognition [93]. A Maximum Margin Criterion (MMC) method [14] is designed to guide latent feature extraction, which is also robust to the noise data. With the development of deep learning, learning-based approaches [8, 192, 234, 236] have dominated feature extraction.

These feature extraction approaches depend on the redundant visual information from images or videos. However, event streams lack details of objects, so model-based approaches cannot achieve good results. Some model-free approaches can be further explored for event cameras.

2.2.3 Classification for Gait Recogntion

After extracting the features from the raw data, the classification based on these features decides the gait recognition performance. SVM is a widely used classification method, which can be directly applied to high-dimensional features. Independent Component Analysis (ICA) and Genetic Fuzzy Support Vector Machine (GFSVM)

are combined to process the features for human identification [142]. A decision tree has also been used for classification and achieves more than 90% accuracy [77], while the hidden Markov model [97, 98] and Bayesian classification [21] are also suitable for this task.

Deep learning-based approaches can perform end-to-end gait recognition without a clear boundary to distinguish feature extraction and classification. Some blocks of neural networks play the role of a classifier. CNN architecture with multilayer perceptron is a fundamental structure for computer vision tasks, and it has also been used for gait recognition. A neural network, which consists of three convolutional blocks (including convolutional layer, non-linear activation layer, normalization layer and pooling layer) and MLP, are designed for gait recognition directly using GEI [240], and meanwhile, GEINet uses two similar convolutional components and two fully connected layers [192]. 3DCNN is an extension of CNN, which performs convolution on three-dimensional space. The 3D convolution can effectively capture the spatiotemporal features from the gait sequence [127, 234]. Apart from the cross-entropy loss function that is employed for the aforementioned neural networks, contrastive loss [246], triplet loss [207, 214], Siamese loss [198], quintuplet loss [248], centre loss [122] and centre-ranked loss [203] are also utilised for training.

Because gait can be considered as a sequence of movement, recurrent neural network (RNN) architecture can be adopted for recognition as well. The skeleton-based RNN approach performs the recognition using the temporal relationships of joints [128] or between the partial features [20]. Generative Adversarial Networks (GANs) can generate identically distributed synthesised data to form a robust classifier. Multi-task GAN (MGAN) has been designed to recognise multi-view gait [83]. Then Two-Stream GAN (TS-GAN) [229] synthesises GEIs from different angles of view and the corresponding global and partial features. Capsule Network [238] and GCN [118] have also been adopted for gait recognition.

A type of neural networks can effectively abstract a type of features, but various types of features may be related to gait recognition. A lot of hybrid networks combine more than one type of network to extract more valuable features. The combination of CNN and RNN is a widely used structure. CNN is able to generate the spatial

features from raw images or silhouettes, while RNN can capture some temporal features [251]. The ResNet, a kind of CNN, and LSTM are packed for irregular gait recognition [119]. This combined structure works for skeleton-based recognition tasks as well [10]. Auto-encoder (AE) is the top half of auto encoder-decoder, which automatically encodes the input as low-dimensional features and restore the features to the original input. The middle outputs, low-dimensional features, are treated as the encoding of the input. Some new neural networks that combine AE, RNN and GAN also have some achievements in gait recognition. For the combination of AE and RNN, AE firstly encodes the segmented images that only include the individuals, and then LSTM works for classification [252, 253]. GaitGAN [244] and the Alpha-blending GAN (Ab-GAN) [121] utilise AE in GAN to encode the features. These classification approaches also can be employed for event cameras after extracted event-based gait features.

2.2.4 Datasets for Gait Recogition

The development of gait recognition algorithms is benefiting from the increasing number of datasets. The scale of subjects increases from tens to thousands, and scenarios vary from simple to challenging. Different angles of view, clothes, carries, light conditions make the captured data more similar to the real application conditions. The performance of new algorithms keeps growing in various complex environments. Some commonly used datasets are summarised as follows:

- **CMU Motion of Body (MoBo) Dataset [75].** CMU MoBo dataset is an early large scale gait dataset, which is released in 2001. MoBo contains 25 individuals performing four different walking styles (slow walk, fast walk, incline walk and walking with a ball) on a treadmill. Six cameras are evenly distributed around the treadmill, whose resolution is 640×480. Each camera generated a video of about 11 seconds, and more than 8000 images were captured for each individual. This dataset provides both raw RGB images and the corresponding silhouettes. As the earliest available dataset, some approaches [48, 116, 134, 234] are evaluated based on this dataset.

- **USF Human ID Gait Challenge (HumanID Gait) Dataset [185].** HumanID gait dataset is captured for the HumanID gait challenge problem, consisting of 1870 sequences from 122 individuals. This data is collected under five different conditions, and there are various angles of view, shoe types, walking surfaces, elapsed times and carries. The conditions can be mixed and generate up to 32 different combinations. As the complex scenarios in this dataset, some robust algorithms [78, 135, 211] are dedicated for one of these scenarios and evaluated using this dataset.

- **CASIA Gait Database: Dataset B (CASIA-B) [243].** The CASIA-B dataset, the second gait dataset released by CASIA, is the most widely used for gait recognition until now. The first CASIA gait dataset, CASIA-A, only contains 20 subjects from three cameras that viewing angels are 0°, 45° and 90°, respectively. CASIA-B is larger than CASIA-A, which consists of 124 individuals, and the 11 cameras are evenly deployed with the viewing angles from 0° increasing to 180°. In addition to the normal walking style, two additional conditions, walking with a coat and walking with a bag, are involved in this dataset. Until now, the state-of-the-art neural network approach, 3DCNNGait [127], can achieve more than 90% accuracy with CASIA-B.

- **CASIA Infrared Night Gait Dataset (CASIA-C) [209].** The CASIA-C dataset includes night's gait data captured by the thermal infrared camera. 153 individuals are involved in this dataset, who walk at different speeds (slow, normal and fast). The condition of whether individuals carry a bag is also considered in this dataset. A combination of CNN and RNN is employed for gait recognition for this dataset [19].

- **Osaka University and the Institute of Scientific and Industrial Research Large Population (OU-ISIR LP) Dataset [90].** The OU-ISIR LP dataset is extremely large-scale, including 4007 individuals. There are 2135 males and 1872 females with ages from 1 to 94 years old, and cameras with multi-views are also employed. In addition to the general large population (OU-ISIR LP) dataset, treadmill dataset with various speeds and clothes (OU-ISIR Treadmill) and large

population dataset with bags (OU-ISIR LP-Bag) are additionally released. With the large volume of this dataset, more and more algorithms [133, 237, 248] utilise it as the benchmark for evaluation.

- **TUM Gait from Audio, Image and Depth (GAID) dataset [85].** The TUM GAID dataset provides simultaneous RGB images, depth images and audio streams of individuals, which extends gait recognition from a single data modality task to a multi-modality task. This dataset contains 32 subjects and involves three different conditions (time, carries and shoes). A Microsoft Kinect sensor is employed during collection and outputs a video stream with 640×480 resolution and 30 FPS frame rate. A depth-based histogram energy image [84] was designed to solve this problem, and the traditional RGB images are also used for general comparison [120].

These gait datasets increasingly broaden the application scenarios of RGB cameras for recognition. For event cameras, the dedicated gait dataset should be prepared for both training and evaluation. On the other hand, a dataset that includes both RGB images and event stream is also required to compare the recognition performance using these two modalities.

2.3 Visual Privacy Protection and Encryption

2.3.1 Encryption-based Visual Privacy Protection

Privacy is a crucial challenge for all kinds of vision sensors. Prior to the presence of event cameras, many encryption algorithms had been designed for streaming data, such as Data Encryption Standard (DES), International Data Encryption Algorithm (IDEA) and Advanced Encryption Standard (AES). However, these algorithms are inefficient for vision sensors, since visual data is redundant and the scale is large. Specific encryption schemes have been designed for various vision sensors.

The Arnold Cat Map is a basic algorithm for pixel scrambling in the space domain for encrypting images [76]. It applies matrix transformation to scramble adjacent pixels rapidly. Several encryption algorithms have extended Arnold Cat Map to

achieve better performance. Furthermore, chaotic-system-based encryption algorithms employ the pseudorandom sequence to encrypt the values of pixels, such as Logistic Mapping [169] and Chebychev Mapping [88]. Besides, encryption schemes in the transform domain [212, 254] and partial encryption schemes [204] are also used for image encryption. Nevertheless, these schemes cannot be directly applied for event cameras, since their outputs are sparse in the space domain, which are different from images.

Video has both streaming data and image features, and thus video encryption should be designed with considerations on both security and efficiency. VEA [174] directly applies streaming data encryption, i.e. DES, to encrypt video stream. Compared with image encryption, this approach increases the efficiency of encryption and the security is preserved. CSC [45] utilizes three chaotic mapping algorithms to generate a chaotic sequence, and an XOR operation is involved in generating the ciphertext. Moreover, some video encryption algorithms take advantage of the video's encoding format to encrypt the key information, such as the widely-used MSE [231] and MHT [235]. These video encryption schemes rely on the encryption of critical information in a single frame or adjacent frames. However, such approach is not suitable for event cameras either, because each event cameras' output holds a small amount of information only.

The point cloud is a new kind of visual data representing a 3D structure of an object. Some extended chaotic mapping algorithms [92, 95] have been used to encrypt the point cloud. Furthermore, a series of random permutations and rotations have been employed to encrypt the point cloud by deforming the geometry [96]. One intuitive method is processing a stream of events as a point cloud, due to the structural similarity between them. However, the minimum and maximum values of the encrypted point cloud in each dimension are not consistent with the original point cloud, which prevents the application of these encryption schemes for event cameras.

2.3.2 Other Visual Privacy Protection Approaches

Encryption is a widely used approach to protect all kinds of data, and non-encrypted privacy protection tends to make use of some characteristics of devices and data.

For visual data, downgrading the quality of captured images is possible to protect the related privacy. However, another problem is how to balance privacy and the performance of downstream tasks because the degraded image generally cannot be restored to its original quality. In some sensitive areas, a pulsing light, which distorts imagery, is produced to prevent the unauthenticated camera[170]. A thin film organic polymer is also utilised for the lens to block imagery[149]. These approaches focus on the environments and devices, which thoroughly destroy the captured image. More non-encrypted approaches destroy the sensitive part of the image and keep other information that can be used for other tasks that are not privacy-related.

Facial features are seriously sensitive information, which has been involved in several authentication systems. If the captured images are not used for face recognition or other face-related tasks, the face information is sensitive and should be protected. The K-same algorithm [157] replaces the original face with the average k face images, efficiently preventing face identification. FaceSwapping [30] directly replaces the face in the image with a similar image in the large-scale face image library collected from the internet. The above approaches only solve face privacy, and some removal, abstraction and replacement approaches can be used for general objects or people. Some image or video inpainting approaches, such as quilting [64] and exemplar-based methods [51], can generate the synthesized texture to compensate for the removed sensitive part.

In addition to destroying or hiding the privacy-related part of images, they can be described as high-dimensional features for particular tasks. Some privacy-protection approaches convert the raw images to some secured features which do not include visual privacy. A secure multi-party technique is involved for vision-based classification, including an oblivious transfer operation, a secure dot-production and secure Millionaire protocols [12]. Besides, a secure fuzzy matching method based on converted attributes is also employed for image matching [13]. In addition, Shashank et al. have proposed a hierarchical index structure and a hash-based indexing scheme to retrieve similar images from the data without exposing the visual information of the query images [189].

These privacy protection approaches fully utilise the characteristics of images

and destroy the recognisable information. However, the original event streams are not directly recognisable, which can be converted to meaningful images and videos using some reconstruction algorithms. Thus, the balance between the security and the application of event cameras can be further explored.

Chapter 3

EV-Gait: Bringing Gait Recognition from RGB Cameras to Event Cameras

3.1 Introduction

Gait recognition is a fundamental building block in many real-world applications such as activity tracking, digital healthcare and security surveillance, which aims to recognise human identities based on their walking patterns captured by the sensors. A variety of sensors have been employed for gait recognition, such as standard (RGB-based) cameras, infrared cameras, floor sensors and inertial sensors. Event cameras are a new kind of vision sensor, which have unique advantages over the standard RGB cameras for gait recognition because: (i) their low energy and bandwidth footprint make them ideal for always-on wireless monitoring; and (ii) the high dynamic range allows them to work under challenging lighting conditions without dedicated illumination control.

However, event cameras operate in a completely different way compared with RGB cameras, which generate asynchronous and noisy events rather than frames when capturing human motions. As presented in Fig. 3.1, the left figure shows the RGB gait images captured by standard cameras, while the right figure demonstrates the gait

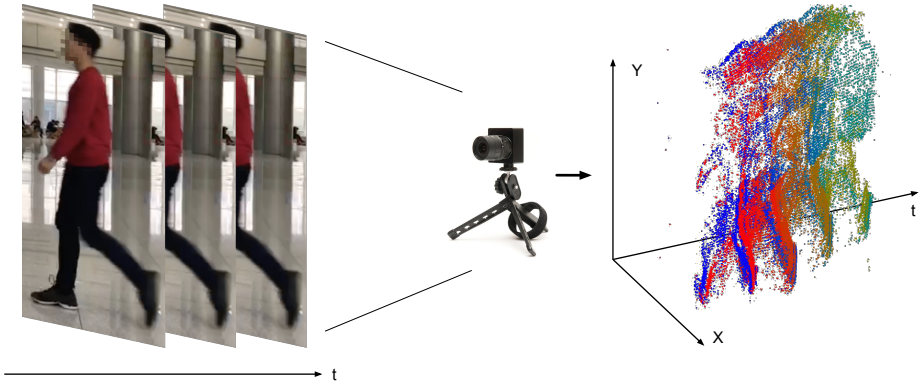


Figure 3.1: Comparison of the outputs between standard cameras and event cameras

event stream generated by event cameras. For the event stream, the positive intensity changes (+1) are denoted in red and the negative intensity changes (-1) are denoted in blue, and the contrast decreases following the passage of time. Because of the different structures of images and event streams, the conventional RGB-based image processing and gait recognition approaches cannot be directly applied on the event data.

In this chapter, we investigate the feasibility of using event cameras to tackle the classic gait recognition problem.

- No event-based gait dataset is available for training and evaluation.
- No dedicated gait recognition approach has been designed for event stream.

In order to solve the aforementioned problems, we firstly produced two event-based gait datasets, which can be used for training neural networks and further quantitative evaluation. Then, we propose a new event-based gait recognition approach, namely EV-Gait, which is able to work with the noisy event streams and accurately infer the identities with gait recognition. Lastly, extensive experiments are conducted to analyse the proposed approach. Concretely, major contributions in this chapter include:

- We propose a novel event-based gait recognition approach EV-Gait, which is specifically designed for event cameras. It is able to effectively remove noise in the event streams by enforcing motion consistency, and employs a deep neural network to recognise gait from the asynchronous and sparse event data.

- We collect two event-based gait datasets DVS128-Gait and EV-CASIA-B from both real-world experiments and public gait benchmarks. DVS128-Gait has been utilised to train the neural network.
- Evaluations based on the two datasets show that the proposed EV-gait can recognise identities with up to 96% accuracy in real-world settings, and achieve comparable (even better in some angles of view) performance with the state-of-the-art RGB-based approaches.

The rest of this chapter is organised as follows. In Section 3.2, we present two image-like representation approaches for event streams, namely count image and time surface, which are concatenated and utilised as the input of our proposed neural network. In Section 3.3, a noise cancellation method is designed to mitigate the effect of noisy events on gait recognition. In Section3.4, EV-Gait is proposed, which includes a feature extraction network and a classification network. The collected event-based datasets and experimental results based on the datasets are presented in Section3.6, followed by a summary concluded in Section3.7.

3.2 Image-like Representation

Unlike conventional RGB cameras, event cameras produce asynchronous event streams that cannot directly fit the CNN-based model. In order to utilise the existing computer vision technologies, we choose to convert the event stream to an image-like representation. Because each event holds temporal, positional and visual information, the converted representation can accumulate such information. EV-FlowNet [255] is a self-supervised deep learning approach to estimate the optical flow for an event stream, and its input is the processed event stream. Although the output of EV-FlowNet is the optical flow whose size is different from that of the gait recognition task, its en-coder part has inspired the design of the counterpart in our gait recognition framework, because both of them project the events into another space as a feature matrix. Inspired by this work, the number of events that happened at each pixel and the time when events happened are employed as the representation approach to describe an

Figure 3.2: Visualisation of the CI^+ in 50ms

Figure 3.3: Visualisation of the CI^- in 50ms

event stream. Concretely, the count image accommodates the counts of positive or negative events at each pixel, respectively, which can effectively describe the spatial characteristics of the event stream, and the time surface holds the time ratios describing the temporal characteristics.

3.2.1 Count Image

An event implies that an intensity change reaches or exceeds the threshold at one pixel, and thus, the number of events that happened at the same pixel is related to its corresponding total intensity change in duration. The thresholds for positive and negative events can be different, so events with different polarities are accumulated, respectively. This count image is formulated as,

$$\begin{cases} CI^+(x,y) = \sum\limits_{t_i \in T, p_i = +1} \delta(x - x_i, y - y_i) \\ CI^-(x,y) = \sum\limits_{t_i \in T, p_i = -1} \delta(x - x_i, y - y_i) \end{cases} \tag{3.1}$$

where an event $e_i = (t_i, x_i, y_i, p_i)$, T is the time duration of the event stream and

Figure 3.4: Visualisation of the TS^{+} in 50ms

Figure 3.5: Visualisation of the TS^{-} in 50ms

δ is the Kronecker delta function. CI^{+} and CI^{-} are the two channels of the count image. Some sample CI^{+} of event streams in the collected event-based gait dataset are visualised as Figure 3.2, and their corresponding CI^{-} are illustrated in Figure 3.3. As can be seen from the figures, the count images with different polarities focus on the changes in different directions. CI^{+} describes the changes on the front side, while CI^{-} presents the changes on the backside. Combined with these two channels, the count image can be utilised to describe the spatial characteristics of an event stream. However, this representation format ignores the temporal features of the event stream. Additional representation should be used to extract the temporal characteristics.

3.2.2 Time Surface

Similar to the count image, the time surface also converts an event stream to an image-like representation. Nevertheless, the time surface focuses on the last active events on each pixel rather than the count. This surface encodes the temporal information from the event stream, which is defined as,

$$\begin{cases} TS^{+}(x,y) = \dfrac{t^{+}_{x,y} - t_{begin}}{t_{end} - t_{begin}} \\ TS^{-}(x,y) = \dfrac{t^{-}_{x,y} - t_{begin}}{t_{end} - t_{begin}} \end{cases} \tag{3.2}$$

where $t^{+}_{x,y}$ and $t^{-}_{x,y}$ are the timestamps of the last positive and negative events at pixel (x, y), respectively, t_{begin} is the timestamp of the first event and t_{end} is the last event of the whole stream. These surfaces estimate the lifetime of object of interest at different locations. The corresponding TS^{+} and TS^{-} of Figure 3.2 and Figure 3.3 are visualised in Figure 3.4 and Figure 3.5, respectively. Compared with the count images, time surfaces show gradients that follows the passage of time, which represents temporal information of the event stream.

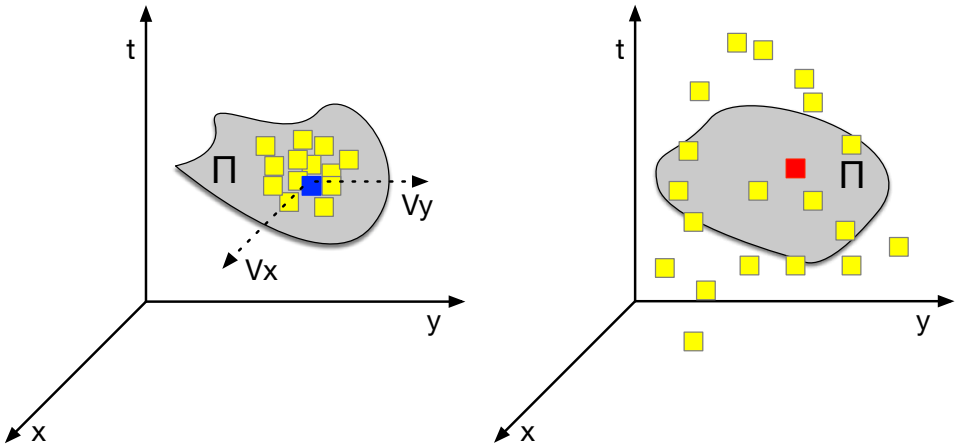
3.3 Noise Cancellation for Event Streams

Event cameras are more sensitive to the intensity change, and thus their outputs are noisier and more evident than traditional cameras. The quality of data heavily affects the performance of gait recognition. Noise cancellation can improve the quality of data as well as the algorithms' performance. In the context of gait recognition, we are only interested in the people walking (or generally objects moving) within the camera field of view, while the other information captured are considered as noise. For event cameras, such noise in the event streams are often cause by the subtle illumination changes in the background, or the unstable nature of the electronic circuits. Therefore, the key challenge of noise cancellation is how we can distinguish if an event is triggered by the moving people/objects of interest or not. This is not a straightforward task, since an event stream spans over both spatial and temporal axis and noise can appear arbitrarily. Most of the existing approaches (e.g. [99, 129, 165]) rely on the simple assumption that the noise in the event streams are ad-hoc and sparse, i.e. they should appear in a random fashion and isolated from the events caused by object motion. These noise cancellation approaches for events are state-of-the-art. Liu et al [129] discard an event as noise if there is no other event captured at its eight neighbour pixels within a certain time period. Khoda et al [99] improve Liu's approach

by recov-ering events that are mistakenly classified as noise. Padala et al [165] filter noise in the event stream by exploiting the fact that two events are fired at the same location can't be too close in the time domain. These approaches only consider the basic generation mechanism of events, but the proposed approach fully employs the characteristics of gait and the speed consistency. However this is not always true, because when the overall lighting condition is not stable, the amount of noise may dominate the stream and bury the events of interest.

To overcome this problem, we consider a new noise cancellation approach by exploiting the motion consistency within the event streams. The intuition is that if an event is caused by the genuine motion of the objects (human body in our gait recognition case), in the near future there should be another events appearing at locations that are consistent with the object motion. In other words, within a local region, the events caused by object motion should be able to form a consistent "moving plane" in the spatiotemporal domain, while the noise event should not. Figure 3.6 demonstrates an example of this idea. We see that in Figure 3.6 (a), for a valid event (the blue dot), there should be a number of previous events that fired in its close vicinity (the yellow dots), since they are triggered by the motion of object across both space and time. Therefore, these events should be able to modelled as a consistent plane Π with velocity (v_x, v_y). On the other hand, as shown in Figure 3.6 (b), if an event is noise (the red dot), the recently appeared events (the yellow dots) typically have no or little spatial correlation, i.e. they can not be described as a consistent plane. In our approach, we exploit this property by looking at the optical flow within the event streams [25], which can naturally assess motion consistency. Event-based visual flow [25] calculates the dense optical flow of an event stream by fitting a surface defined by the neighbour events. This optical flow is related to the moving speed of objects in the real world. Because the speed of gait is in a limited range, the noise can be removed accord-ing to the estimated speed. The proposed noise cancellation approach is inspired by this surface fitting approach to estimate the speed and further remove the noise.

Concretely, to compute the optical flow of an event e_i, we drop its polarity, and express it in the three dimensional space as $e_i = (t_i, x_i, y_i)$. Then the plane where e_i



(a) Valid events (b) Noise events

Figure 3.6: An example of our noise cancellation approach based on motion consistency

is on can be described as

$$ax_i + by_i + ct_i + d = 0 \tag{3.3}$$

where a unique $(a, b, c, d) \in \mathbb{R}^4$ defines a unique plane Π.

The position for those events that are within close proximity of e_i in both spatial and temporal axis, we fit a plane via least squares:

$$\hat{\Pi} = \underset{\Pi \in \mathbb{R}^4}{argmin} \sum_{j \in \mathcal{S}_i} \left| \Pi^T \begin{pmatrix} x_j \\ y_j \\ t_j \\ 1 \end{pmatrix} \right|^2 \tag{3.4}$$

where $\mathcal{S}_i$ is the event set including both e_i and the events appear within the 3×3 neighbourhood of (x_i, y_i), and the time window $[t_i - \Delta_t, t_i + \Delta_t]$. In our experiments we set Δ_t to 1ms.

Let us assume that a unique plane $\hat{\Pi}\left(\hat{a}, \hat{b}, \hat{c}, \hat{d}\right)$ is obtained. Then we calculate its velocity at the event e_i as:

$$v = \begin{bmatrix} v_i^x \\ v_i^y \end{bmatrix} = -\hat{c} \begin{bmatrix} \frac{1}{\hat{a}} \\ \frac{1}{\hat{b}} \end{bmatrix} \tag{3.5}$$

where v_i^x and v_i^y are the velocity of event e_i along the x and y axes respectively.

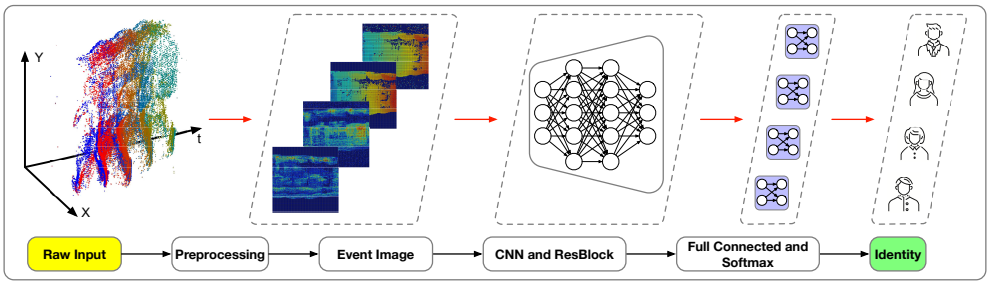


Figure 3.7: Workflow of the proposed EV-Gait.

Then we validate the motion consistency by checking the velocity v. If $0 < |v| < V_{max}$, we accept e_i, since a valid event caused by genuine motion should be moving, and the speed should be within certain reasonable range. Otherwise, we declare e_i as noise, and remove it from the event stream. We do this iteratively for each event until all the events in the stream are considered as valid.

3.4 EV-Gait: Event-based Gait Recognition

EV-Gait, as shown in Figure 3.7, starts from capturing asynchronous raw event stream while the subject is walking through the view. Then the raw event stream is preprocessed through event noise cancellation and represented according to the design of the input layer of the deep neural network for gait recognition. At last, we train our deep network and apply it to recognise the identities of the subjects based on event streams.

Our deep neural network for event-based gait recognition can be vastly divided into two major components: convolutional layers with Residual Block (ResBlock) are responsible for feature extraction and fully-connected layers with softmax associate the features to different identities. The convolutional layers have been proved an effective way to extract features and popularly applied in image classification tasks [72, 107, 178]. The ResBlock layers [81] are able to deal with the vanishing features problem when the network goes deeper so that features extracted by convolutional layers can be better integrated. The fully-connected layers decode the features and pass them to the softmax functions to execute classification tasks.

3.4.1 Feature Extraction Network

The structure of the feature extraction network is illustrated in Figure 3.8. Because the resolutions of event cameras are relatively small, e.g, 128×128 for DVS 128, the vanilla convolutional layers are firstly utilized to extract low-level features. Meanwhile, Compared with the larger convolutional kernels (7×7 or 5×5), the feature extraction networks for events is only equipped with 3×3 kernels. The desired features for gait recognition are the walkers' skeletons and their walking patterns rather than their clothing and other details. The number of convolutional layers in the feature extraction network associates with this target, and only four downsampling convolutional layers are employed in the first part. Their strides are two, and the numbers of output channels are 64, 128, 256 and 512, respectively. An activation function, ReLU [156], follows each convolutional layer. Here, we do not use any off-the-shelf RGB feature extraction network with pre-trained parameters, such as LeNet, AlexNet and VGG, since event images do not have image-like fine-grained details. Overly focusing on the details of event images may lead to overfitting due to the noise and irrelevant events.

After passing through the four vanilla convolutional layers, two ResBlocks are employed to enhance the high-level features, which structure is shown in Figure 3.9. The input shape of the ResBlocks is 8×8×512, and the output is the same shape. A ResBlock consist of two convolutional layers, two batch normalisation layers and one activation function, which has two paths to directly passing the input and calculating the residuals, respectively. For the calculating residuals path, the input goes through the first convolutional layer with 3×3 kernels, whose stride is one, and the number of output channels is still 512. The first batch normalisation layer and the ReLU activation function follow the first convolutional layer. The second convolutional layer is the same as the first convolutional layer. After this convolutional layer, the original input, from the direct path, adds the residuals from the second batch normalisation layer. The features generated after two ResBlocks are passed to the classification network for the final recognition.

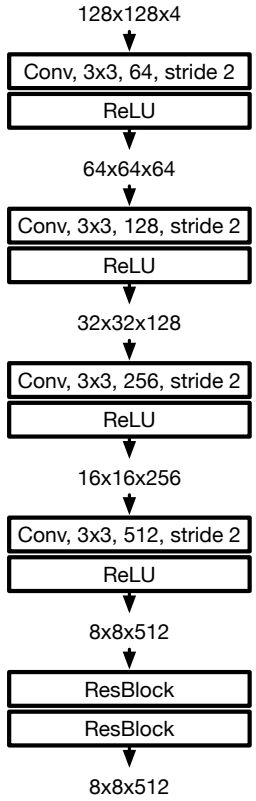


Figure 3.8: The super-parameters and architecture of the feature extraction network

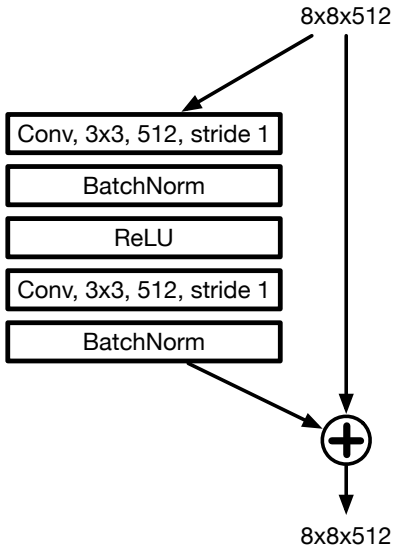


Figure 3.9: The ResBlock structure

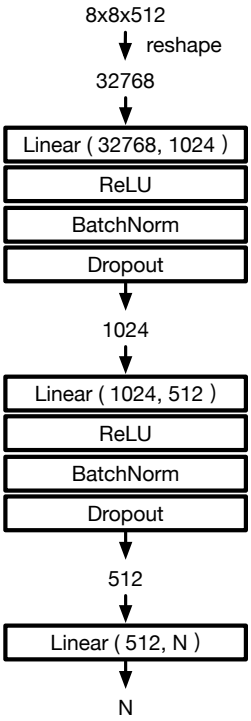


Figure 3.10: The structure of the classification network

3.4.2 Classification Network and Loss Function

A Multi-Layer Perceptron (MLP) is employed as the classifier of the EV-Gait, which includes two hidden layers. The output shape of feature extraction network is $8 \times 8 \times 512$, which is flatted as a vector, which length is 32768. The first hidden layer has 1024 units, and is followed by an activation function (ReLU), a batch normalisation layer and a dropout layer. There are 512 hidden units in the second layer, and except for the batch normalisation layer, the rest structure is the same as the first one. A final linear layer maps 512 hidden units to the N identities, and the length of the final output is the number of identities that need to be recognised. The structure of the classification network is illustrated in Figure 3.10.

The output of the classification network is converted to the probabilities of different identities using the softmax function, which is defined as follows:

$$p_i = \frac{exp(c_i)}{\sum_{i=1}^{N} exp(c_i)} \tag{3.6}$$

where p_i is the probability of the i-th identity, c_i is the i-th element of the classification

network output. At last, the cross entropy loss function (Equation 3.7) and Adam optimizer [102] are adopted to train the network.

$$Loss = -\sum_{i=1}^{N} y_i \log(p_i) \tag{3.7}$$

where y_i is the true label of the input in one-hot format and p_i is the predicted probability, and the prediction is to choose the result with the highest probability:

$$\hat{y} = \arg\max_i(p_i) \tag{3.8}$$

3.5 Datasets

The event camera is an emerging vision sensor; thus, there is a limited number of datasets for specific tasks such as gait recognition. Two datasets, DVS128-Gait and EV-CASIA-B, are presented. They are captured by an event camera, which serves as a basis for model training as well as quantitative evaluations and comparisons. DVS128-Gait is captured in real-world settings in a lobby of a teaching building and volunteers vertically walk in the front of the camera, similar to the real application scenarios. EV-CASIA-B is generated from the widely used RBG gait benchmark, CASIA-B, with an event camera that captures the monitor when playing the videos of CASIA-B.

3.5.1 DVS128-Gait Dataset

The DVS128-Gait dataset is collected in real-world settings with a cohort of 21 volunteers over three weeks in a lobby of a teaching building. 15 males and 6 females are recruited to contribute their data in two experiment sessions spanning over three weeks time. In each session, the participants were asked to walk normally in front of a DVS128 sensor mounted on a tripod, and repeat walking for 100 times. The sensor viewing angle is set to approximately 90 degrees with respect to the walking directions. The second experiment session was conducted after a week since the end of first session to include potential variances in the participants gait. Therefore,

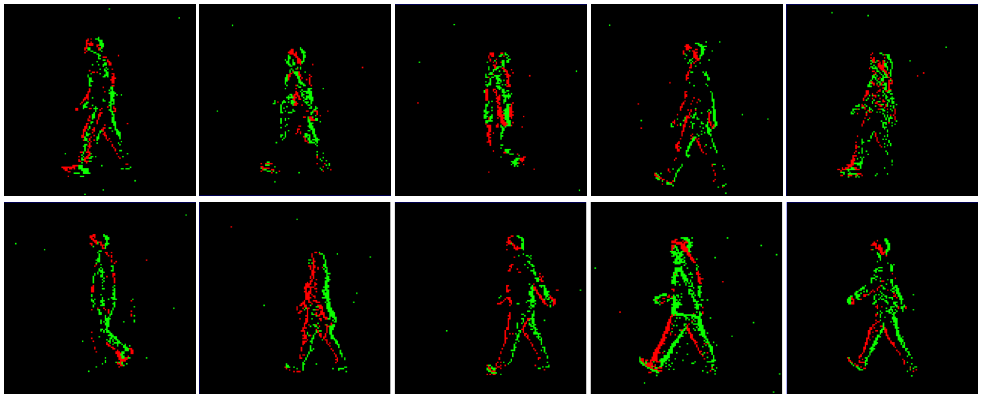


Figure 3.11: Visualisation of the event streams (accumulated over 20ms) of 10 different identities in the DVS128-Gait dataset.

in total we collected 4,200 samples of event streams capturing gait of 21 different identities. Figure 3.11 shows visualisation of the data from 10 different identities (events accumulated within 20ms), where the colour of pixels indicate polarity (red for +1, green for -1). Some moving edge of the volunteers can be early recognised from the visualisation. More statistics features of records are shown in Table 3.1. The time lengths of records are from about three seconds to about six seconds, and their numbers of events are from 40,000 to 100,000.

3.5.2 EV-CASIA-B Dataset

Traditional cameras and event cameras provide different modalities of vision information. To compare the advantages and disadvantages of these two modalities, a widely used RGB benchmark, CASIA-B, is converted to an event-based format, EV-CASIA-B. CASIA-B is one of the most popular benchmark for RGB camera-based gait recognition methods [18, 74, 109, 112]. It contains data from 124 subjects, each of which has 66 video clips recorded by RGB camera from 11 different view angles (0° to 180°), i.e., 6 clips for each angle. The view angle is the relative angle between the view of the camera and walking direction of the subjects. To convert the CASIA-B dataset to event format, we use a similar approach as in [86] and use a DVS128 sensor to record the playbacks of the video clips on a screen. In particular, we use a Dell 23 inch monitor with resolution 1920×1080 at 60Hz. Figure 3.12 shows some examples from the original CASIA-B dataset (top row) and the visualisation of the corresponding

	Training Sequence		Validation Sequence	
No.	Time Length	# of Events	Time Length	# of Events
1	5700±2868 ms	51767±3891	5599±2948 ms	50694±6317
2	5129±2427 ms	50932±5148	4007±1462 ms	49185±5824
3	4856±2304 ms	53375±7049	5626±2487 ms	46830±3752
4	4992±2459 ms	61389±7896	4261±816 ms	62323±7845
5	4812±1380 ms	40481±3147	6728±3925 ms	41829±3521
6	5714±1366 ms	60517±4641	5411±1695 ms	59076±3851
7	5675±2395 ms	46746±4702	5664±2400 ms	46746±4702
8	4112±1669 ms	49013±7964	3442±1175 ms	91538±43827
9	3737±1145 ms	51982±3370	4018±1842 ms	81876±38346
10	4644±1475 ms	49506±6990	5131±1707 ms	49427±5764
11	5457±1873 ms	49672±7623	5484±1799 ms	53724±7489
12	4556±1733 ms	55487±3700	3755±1893 ms	50453±11410
13	5765±2729 ms	41402±3149	4134±1542 ms	43897±3067
14	5638±1853 ms	60143±3698	5012±1575 ms	62567±7998
15	4820±1652 ms	53768±3669	4941±1666 ms	54317±5023
16	5030±1918 ms	50770±5803	5178±2010 ms	52618±2962
17	4764±1478 ms	70373±5414	4634±2207 ms	72839±7641
18	4322±1407 ms	61952±8051	3233±950 ms	60701±7503
19	5504±1784 ms	57938±5173	5987±1675 ms	59774±5185
20	4414±1321 ms	47299±5720	3641±1172 ms	41061±4678
21	3645±1492 ms	36117±6829	3895±1013 ms	41197±4223

Table 3.1: The time lengths and the numbers of events of each volunteer's records

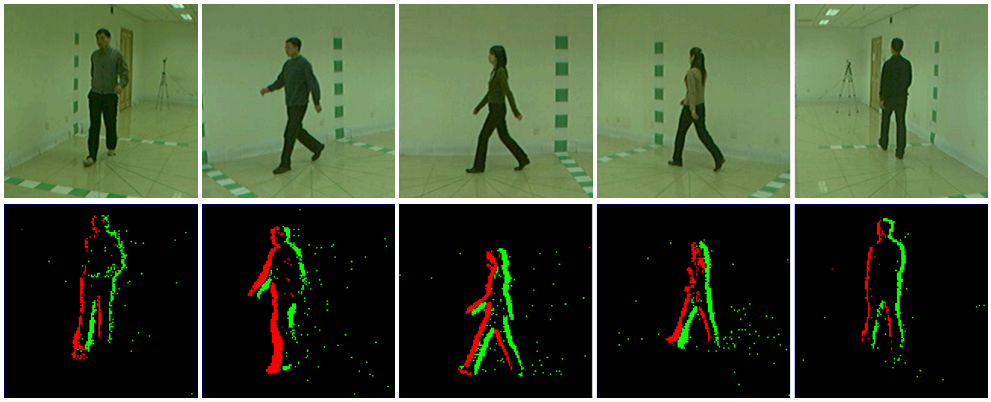


Figure 3.12: The original CASIA-B dataset and visualisation of the corresponding event streams (accumulated over 20ms) in our converted EV-CASIA-B dataset

event streams in our converted EV-CASIA-B dataset (bottom row). The time lengths of records are equal to the original clips, about five seconds, and their numbers of events are from 80,000 to 300,000.

3.6 Evaluation

In this section, we evaluate EV-Gait with both data collected in real-world experiments and converted from publicly available RGB gait databases. In our experiments, we use a DVS128 Dynamic Vision Sensor from iniVation [3] operating at 128×128 pixel resolution. The event data is streamed to and processed on a desktop machine running Ubuntu 16.04, and the deep network (discussed in Section 3.4) is trained on a single NVIDIA 1080Ti GPU. In the following, we first evaluate the performance of event noise cancellation of EV-Gait, and then present the gait recognition performance.

3.6.1 Baselines of Event Noise Cancellation

We compare the proposed noise cancellation technique in EV-Gait against the following three state of the art approaches:

(1) **Liu et al** [129], which discards an event as noise if there is no other event captured at its eight neighbour pixels within a certain time period;

(2) **Khoda et al** [99], which improves Liu's approach by recovering events that are mistakenly classified as noise;

(3) **Padala et al** [165], which filters noise in the event stream by exploiting the fact that two events are fired at the same location can't be too close in time domain.

To fully investigate the noise cancellation performance of EV-Gait, we consider two experiment scenarios, where the event camera is configured to capture: i) a static background with nothing moving; and ii) an artificial object moving upon the background.

3.6.2 Noise Cancellation with Static Background

In this experiment setting, we configure the DVS128 camera to face white walls and continuously capture the event streams for fixed time intervals. The environments are controlled and there is no moving object or shadow within the camera field of view, so that the scene captured by the camera is purely static background. We consider two different lighting sources, i) the light-emitting diode (**LED**) and ii) fluorescent tube light (**FTL**), both of which are AC powered. However, the flicker frequency of the fluorescent light is relatively slow (100Hz or 120Hz), and thus can be easily picked up by the event camera, causing more noise in the event streams. On the other hand, the LED lights used in our experiments are more stable, since they use rectifiers to convert the AC to DC and smooth the output with capacitors. Figure 3.13(a) and Figure 3.14(a) show the the recorded events accumulated within a 20ms window under the two different lighting sources respectively. Clearly in this case, all the events (white dots) should be noise, since the event camera is only capturing the static white wall. We then apply the event noise cancellation technique used in EV-Gait and the competing approaches to the recorded event streams, and Table 3.2 shows their performance in removing noise. The first column shows the total numbers of noise events under the two lighting conditions, while the rest show the percentage of noise events left after applying individual approaches.

Firstly, we find that the amount of noise caused by fluorescent tube light (FTL) is much more than that of the LED light (1,082,840 vs. 19,009 noise events), which confirms that event cameras are very sensitive to different lighting conditions. On the other hand, we see that our technique (Figure 3.13(b) and Figure 3.14(b)) can effectively remove most of the noise in the event streams, up to 97.79% and 99.73%

	# of Noise	EV-Gait	Liu [129]	Khoda [99]	Padala [165]
LED	19,009	**2.21%**	29.3%	5.13%	15.56%
FTL	1,082,840	**0.27%**	48.25%	21.06%	47.37%

Table 3.2: Noise cancellation performance of the proposed and competing approaches under LED and FTL lights.

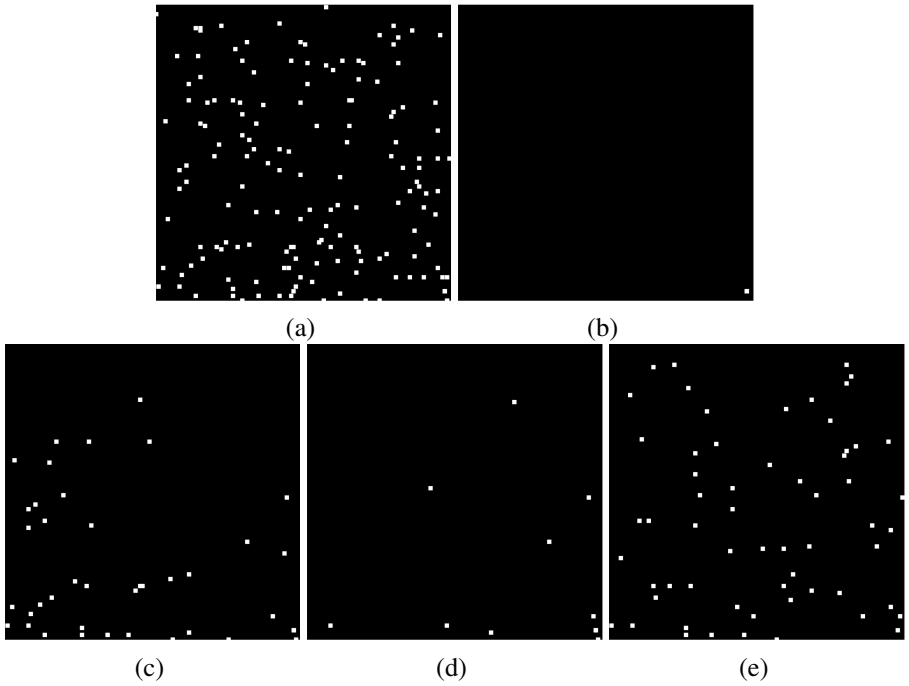


Figure 3.13: Visualisation of events (400ms) captured for a static background under LED lighting

under LED and FTL. This significantly outperforms all the completing approaches (see Figure 3.13 and Figure 3.14 for visualisation of the remaining noise events), where the best one (Figure 3.13(d) and Figure 3.14(d), Khoda [99]) keeps almost 78 times (21.06% v.s. 0.27%) more noise events than ours under the unstable FTL lighting. The performance of Liu [129] is visualised as Figure 3.13(c) and Figure 3.14(c), and that of Padala [165] is as Figure 3.13(e) and Figure 3.14(e). This is expected as the competing approaches only use spatial and temporal inconsistencies to filter out noise events, while the proposed EV-Gait exploits moving surfaces based on optical flow, which is inherently more robust.

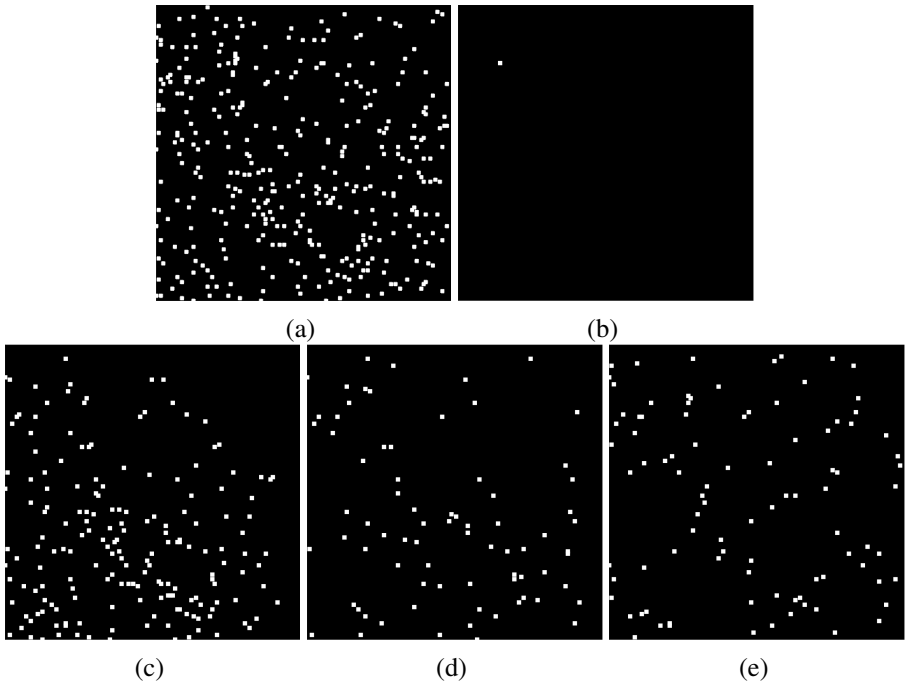


Figure 3.14: Visualisation of events (400ms) captured for a static background under FTL lighting

3.6.3 Noise Cancellation with Moving Objects

The second set of experiments investigate the performance of different noise cancellation approaches in the presence of moving objects. We again configure the event camera to face the white walls in both LED and FTL lighting conditions, but rather than capturing the background in this case we use a red laser pointer to generate a moving dot on the wall. This moving dot can be captured by the event camera as series of events, as well as the noise. Intuitively, an ideal noise cancellation approach should only extract the events corresponding to that moving dot and discard all the others, forming the complete and clean trajectories. Figure 3.15 (a) and Figure 3.16 (a) show the visualisation of events captured by the event camera under LED and FTL lighting. We can see that although there are trajectories visible, the noise events still occupy most of the scene, especially in the FTL case where the lighting source is not very stable (flickering). Figure 3.15 (b)-(e) and Figure 3.16 (b)-(e) show the visualisation of events produced by EV-Gait and the competing approaches under LED and FTL lighting respectively. We see that clearly the proposed EV-Gait performs the best, in the sense that it can reject most of the noise events spread across the scene while

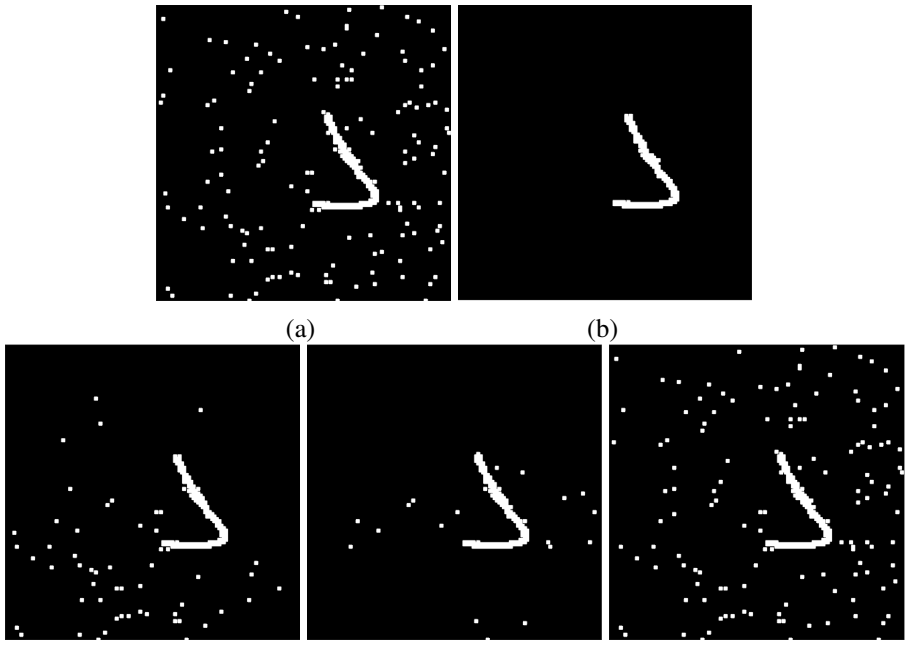


Figure 3.15: Visualisation of events (400ms) captured for a moving object under LED lighting

retaining the positive events corresponding to the moving dot, i.e. preserving the complete and clean trajectories. On the other hand, the competing approaches performs significantly inferior: only Liu [129] and Kohoda [99] could achieve acceptable results under the stable LED lighting (see Figure 3.15 (c)-(d)), but they immediately fail under the unstable FTL condition (see Figure 3.16 (c)-(d)).Padala [165] fails on both LED and FTL conditions (see Figure 3.15 (e) and Figure 3.16 (e)).

3.6.4 Noise Cancellation Sensitivity Evaluation

The event plane is constructed using neighbour events, including actual events and noise events. This is the main limitation of the proposed noise cancellation approach. Because the noise distribution is unknown, we assume that the noise caused by the cir-cuit is relatively uniform, and the noise caused by light is around the actual events. The estimated plane is calculated by minimizing the distance between the plane and a collec-tion of events. The circuit noise is assumed to be uniform, and thus, it has little effect on the estimated plane. The performance of plane estimation is mainly affected by the light noise. In order to further evaluate the performance of the noise

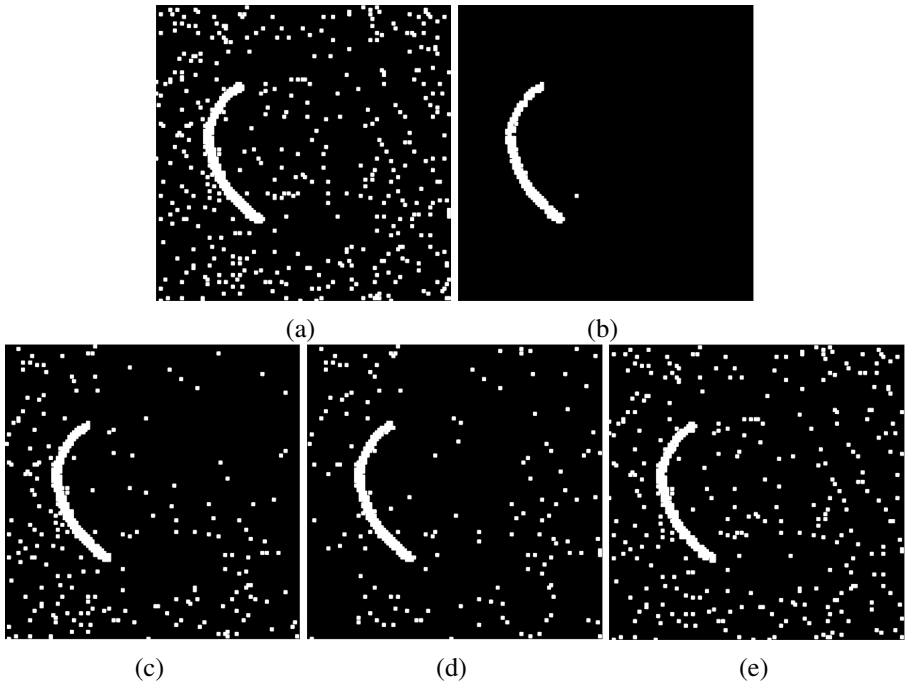


Figure 3.16: Visualisation of events (400ms) captured for a moving object under FTL lighting

cancellation approach, a set of experiments is conducted.

The original event stream with noise is firstly processed using the proposed approach, which produces a list of actual events treated as the ground truth. In order to evaluate the estimated plane and the noise cancellation approach, some new noise events should be mixed into the actual events. In the original event stream, the events which are fil-tered out can be supposed to be noise, and the new noise events are generated by adding some jitter to the timestamp and location of these noise events. Then the generated event stream is processed using the proposed approach again. After the process, the number of actual and noise events is a metric to measure the effect of noise on the plane estima-tion, and the result is shown in Table 3.3

	20%	40%	60%	80%	100%
False Positive	6.28%	7.94%	9.28%	10.74%	12.18%
False Negative	16.40%	18.22%	19.42%	20.78%	22.30%
Overall	22.63%	26.37%	29.00%	31.59%	34.60%

Table 3.3: The sensitivity analysis of the noise cancellation approach

Here, the jitters of timestamp and location are 100 ms and 3 pixels. Different percent-ages of synthesised noise events are mixed to generate the new event stream.

As can be seen, when a small number of noise events (20%) is added, the proposed approach also filters some actual events (about 16.40%) out. More actual events are filtered out with the increase of added noise events. Compared with the actual events filtered out, the percentage of the remaining noise events is low. Only 12.18% noise is left when 100% noise events are added.

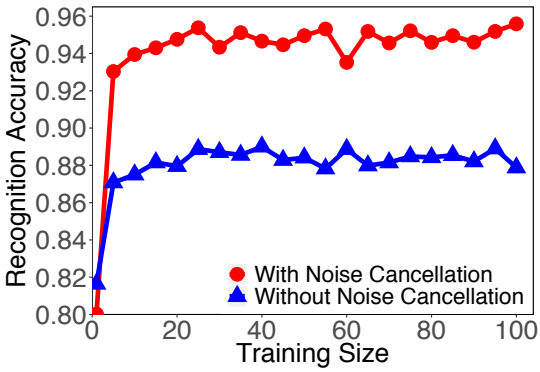
3.6.5 Gait Recognition on the Real-World Scenario

To perform the evaluation on the proposed network, the DVS128-Gait dataset is utilised for training and evaluation.

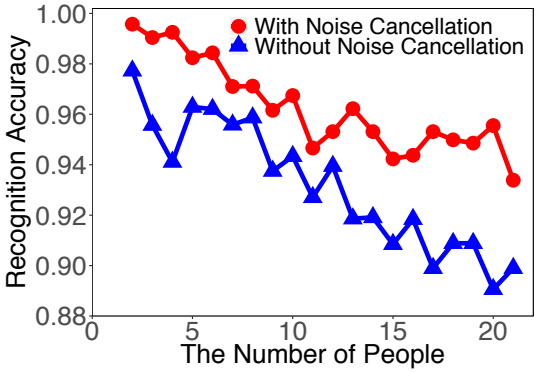
Implementation Details: We implement the proposed deep network in EV-Gait (discussed in Section 3.4) with TensorFlow [5]. The data collected in the first session is used for training, while for testing we use data from the second session. During training we set the batch size as 64 and learning rate as 3e-6. Both training and testing were performed on a 12GB NVIDIA 1080Ti GPU.

Results: The first set of experiments investigate the recognition accuracy of EV-Gait with respect to the amount of training samples per identity. In particular, we use data from all 21 participants, but randomly select different numbers of training samples for each of them, varying from 1 to 100. For each case, we retrain EV-Gait for 30 times and report the averaged recognition accuracy. Figure 3.17(a) shows the results, and we see that as more samples are used in training, the recognition accuracy of EV-Gait increase immediately, while after 25 samples per identity the accuracy tends to be stable (approximately $>$94%). This indicates that EV-Gait doesn't require massive training data to converge, and the recognition accuracy is reasonably good even with data collected from practical settings. On the other hand, we also observe that there is a significant performance gap between using vs. not using the noise cancellation technique, e.g. removing the noise in the event stream using our approach can improve recognition accuracy up to 8%. This confirms that the proposed noise cancellation approach in EV-Gait is crucial, and have very positive knock-on effect on the overall gait recognition performance.

We then study the impact on recognition accuracy when the number of identities considered vary. We randomly select a subset of identities (i.e. participants) in the



(a) recognition accuracy with different training samples per identity



(b)recognition accuracy with different number of identities

Figure 3.17: Recognition accuracy of EV-Gait under different conditions

dataset, from 1 to 21 respectively, and use all samples of the selected identities in the training set (data from the first session) to train EV-Gait. We again retrain the model and report the averaged recognition accuracy over 30 inference on the test set, and Figure 3.17(b) shows the results. We see that as the number of identities increases, the recognition accuracy drops accordingly. This is expected because although we have extra data for training, it is more challenging to distinguish more identities. However, we see that even with 20 identities, EV-Gait can still achieve almost 96% recognition accuracy. In addition, similar with the previous case we observe that the noise cancellation technique in EV-Gait helps a lot, e.g. increasing the accuracy up to 8%.

3.6.6 Gait Recognition on the Synthesis Benchmark

We have showed that EV-Gait performs well in data collected from real-world settings, and now we show that it could also achieve comparable performance with the state-of-

the-art gait recognition approaches that are designed for RGB images. Since those approaches do not work on event streams, for fair comparison, we convert the widely used CASIA-B [243] benchmark into its event version EV-CASIA-B. Then we run EV-Gait on the converted EV-CASIA-B dataset, and compare the resulting recognition accuracy with that of the state-of-the-art approaches on the original CASIA-B dataset.

Implementation Details: We consider the same deep network structure as in the previous experiments on the DVS128-Gait dataset. For training, we use the data of the first 74 subjects to pre-train the network. Then for the other 50 subjects, for each viewing angle we use the first 4 out of 6 clips to fine-tune the network, and the rest 2 clips are used for testing. We implement two competing approaches that work on RGB images: i) **3D-CNN** [236] and ii) **Ensemble-CNN** [236], which can achieve state-of-the-art gait recognition performance on the original CASIA-B benchmark.

Results: Table 3.4 shows the gait recognition accuracy of the proposed EV-Gait with the competing approaches 3D-CNN and Ensemble-CNN. It is worth pointing out that the frame rate of the video clips in CASIA-B dataset is only 25 FPS, with a low resolution at 320×240. As a result when converting such data into event format via playback on the screen, the event camera will inevitably pick up lots of noise. In addition, unlike the original RGB data, the event streams inherently contain much less information (see Figure 3.12). However, as we can see from Table 3.4, the proposed EV-Gait can still achieve comparable gait recognition accuracy (89.9%) with the competing RGB camera based approaches overall (94.1%). For some viewing angles, especially when the walking directions of the subjects are perpendicular with the camera optical axis (e.g. around 90°), the proposed EV-Gait even outperforms the state-of-the-art 3D-CNN and Ensemble-CNN (96.2% vs. 88.3% and 91.5%). This is because in such settings the event streams captured by the event camera can preserve most of the motion features, while removing the gait irrelevant information in RGB images such as cloth texture. On the other hand, for the viewing angles that the subjects walk towards/away from the camera (e.g. 0° or 162°), the accuracy of EV-Gait is slightly inferior to the RGB-based approaches. This is expected, since in those cases compared to RGB images, the event streams contain fewer informative features on the subjects' motion patterns, and thus struggle to extract their identities.

Angle\Methods	3D-CNN	Ensemble-CNN	EV-Gait (Ours)
0°	87.1%	88.7%	77.3%
18°	93.2%	95.1%	89.3%
36°	97.0%	98.2%	94.0%
54°	94.6%	96.4%	91.8%
72°	90.2%	94.1%	92.3%
90°	88.3%	91.5%	**96.2%**
108°	91.1%	93.9%	**91.8%**
126°	93.8%	97.5%	91.8%
144°	96.5%	98.4%	91.4%
162°	96.0%	95.8%	87.8%
180°	85.7%	85.6%	**85.7%**

Table 3.4: Gait recognition accuracy of EV-Gait and two competing RGB based approaches (evaluated on CASIA-B dataset).

3.7 Summary

In this chapter, we have demonstrated the possibility of applying event cameras for gait recognition. After collecting two event-based gait datasets, a neural network is trained based on the datasets, and the result shows that CNN-based approaches are possible to solve gait recognition problems with event cameras. Some takeaway lessons from this chapter are as follows:

- Although each event holds far less information than a single image, the event stream still has enough features for gait recognition. The aggregation approaches for events, such as count image and time surface, can effectively produce an image-like representation for downstream tasks.
- Noisy events in a event stream seriously affect the accuracy of gait recognition with event cameras, and a motion consistency based noise cancellation method can effectively remove the noisy events for gait recognition, which can improve the gait recognition accuracy.
- CNN-based architecture can extract useful features from an image-like representation of an event stream. These features can be used for gait recognition, which achieves comparable results of gait recognition with standard RGB cameras.
- For the multi-view gait recognition problem, the event camera based approach outperforms the state-of-the-art approaches using standard RGB cameras in

some specific angles, probably because gait-irrelevant information, such as cloth texture, is not captured.

In summary, we propose EV-Gait, a novel approach for gait recognition using event cameras in this chapter. EV-Gait features a new event noise cancellation technique exploiting motion consistency of the moving objects to clean up event streams, and can be generally applied on a wide range of applications on tracking, localisation, and activity recognition using event cameras. Then, a deep neural network in EV-Gait is designed for recognising gait from event streams. We collect two event-based gait datasets from both real-world experiments and an RGB-based benchmark. According to the evaluations on the datasets, EV-Gait achieves up to 96% accuracy in real-world settings and comparable performance with state-of-the-art RGB-based approaches on the benchmark.

However, the work in this chapter is an initial step on gait recognition with event cameras, and there are some issues that deserve further exploration:

- Besides the image-like representation, is there any other representation approach that can be used for event stream to recognise persons?
- Apart from the CNN-based architecture, is there any other type of neural network that can extract better features and achieve better results?
- The EV-Gait can perform gait recognition using the whole event stream, and is it possible to conduct gait recognition by using only a partition of event streams? How efficiently can event-based gait recognition achieve that objective?

Chapter 4

3DGraph-Gait: Real-Time Accurate Gait Recognition with Event Cameras

4.1 Introduction

With event-based gait datasets, data-driven neural networks can be trained and quantitatively evaluated. Inspired by traditional computer vision algorithms, a CNN-based approach has been proposed to accomplish gait recognition with event cameras, demonstrating the feasibility of using an event stream to identify persons. However, the characteristics of event streams for gait recognition have not been fully explored and utilised. CNN-based approaches can effectively capture image-like features after packing an event stream into an image-like representation, but spatiotemporal features and local features for event streams have not been considered. On the other hand, the packing-based aggregation method cannot fully utilise event cameras' advantages, such as broad dynamic range and high temporal resolution. These advantages should have the capability of benefiting gait recognition, and the dedicated gait recognition approaches for event cameras should make use of some advantages. Events accumulated in a short period might be enough for gait recognition, but the spatiotemporal features of event stream deserve further exploration.

In this chapter, we propose another representation approach, namely graph-based representation, and develop a corresponding graph-based neural network. Such representation can effectively present spatiotemporal features for an event stream to improve the accuracy of model, and a GCN-based architecture is proposed based on this graph-based representation. Furthermore, we extend the GCN-based architecture from global spatiotemporal feature extraction to local spatiotemporal feature extraction, to support gait recognition with a short period of events only. Extensive experiments are also performed to evaluate the representation approach and the GCN-based model for both the entire event stream and the partial event stream (with events in a short period). Major contributions in this chapter include:

- We explore a 3D graph representation for an event stream, which fully models events' spatiotemporal characteristics. The construction of such a 3D graph includes mapping, sampling and edge generation. Because the constructed 3D graph is further utilised for gait recognition, the remaining spatiotemporal features should be distinct for each identity. Compared with previous image-like representation, this representation is more effective for gait recognition.

- We proposed 3DGraph-Gait, a GCN-based architecture, to extract the spatiotemporal features from the constructed 3D graph. This network includes GMM-based graph convolutional layers, graph residual blocks, graph node clustering and max-pooling layers. After extracting features, some fully connected layers are employed for classification, and the accuracy of this GCN-based approach is higher than the previously proposed EV-Gait.

- We further extend 3DGraph-Gait for supporting events in a short period (about 5-8 milliseconds). Considering the sparsity of such events, a new graph construction method is designed accordingly, and an ensemble mechanism is employed to combine GCNs that deal with multi-time scale features. This architecture can accomplish gait recognition only using events accumulated in several milliseconds, and the ensembled network can achieve higher accuracy.

The background, problems and contributions of this chapter are introduced in this

section. In Section 4.2, we present a graph-based representation approach for event streams, which projects the events to a three-dimensional space. The distances of space and time are the constrains to generate the edges. In Section 4.3, two different sampling methods are described, which are employed for the whole event stream, because the network cannot deal with such a huge number of events. In Section 4.4, we propose 3DGraph-Gait and explain each component of this neural network in details. In Section 4.5, we extend this neural network for supporting events in a short period, and the corresponding ensemble network is presented as well. Evaluations on 3DGraph-Gait and comparisons are demonstrated in Section 4.6, and a summary is made in Section 4.7.

4.2 Graph-based Representation

Compared with the representation approaches described in Section 3.2, the graph-based representation focuses on the spatiotemporal structure of the event stream rather than the image-like visual features. Because an event includes time t, and position (x, y), the outline and the walking trajectory of identity can be recognised by projecting the information into a 3D space, visualised as Figure 4.1. This visualised structure is similar to the point cloud, and the difference is that the third dimension of the event stream is time rather than depth. Thus, some feature extraction approaches for the point cloud can be used for the event stream. The extracted features are spatial features for the point cloud, while spatiotemporal features for the event stream. The graph neural networks have shown their ability in processing point clouds. Because the distance between each point pair among the point cloud represents their spatial distance, individual points can be linked according to their distance to construct a graph. If the distance is less than a given value, an edge will be added for these two points; otherwise, there is no edge. Here, we use a similar approach to construct a graph based on the unlinked events.

An event-based 3D-Graph is constructed by connecting neighboring event nodes with bi-directional edges. Two event nodes $v_i = (x_i, y_i, t_i, p_i)$ and $v_j = (x_j, y_j, t_j, p_j)$

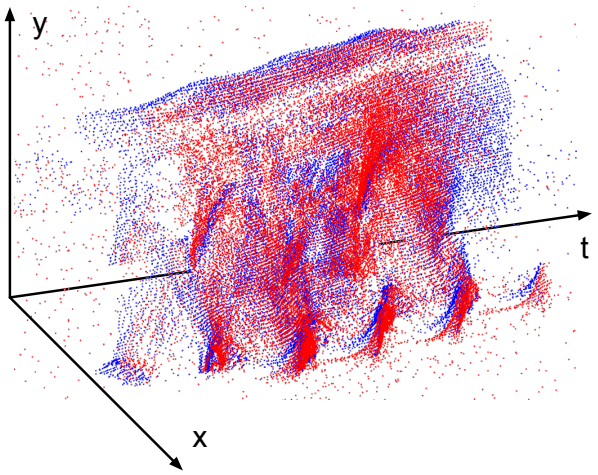


Figure 4.1: Projecting gait events into a 3D space.

are neighbors if their predefined distance is less than the threshold of radius R:

$$\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \alpha(t_i - t_j)^2} < R \tag{4.1}$$

where α is a scaling factor to tune the difference between temporal and spatial resolution of the event-streams. A connected 3D-Graph is represented as $G = (V, E, P)$ where V are the set of vertices and E are the set of the edges. The set of the polarity P are regarded as the input feature set for the graph-based convolution. After the connectivity of the 3D-Graph is determined, the adjacency matrix A of the graph can be generated whose element $A_{i,j}$ equals to 1 if nodes v_i and v_j are connected otherwise it equals to 0. An example 3D-Graph is illustrated in Figure 4.2, where three dimensions are x, y and t. The elements on the diagonal of the adjacency matrix are also set to 1s to include the features of the center nodes when aggregating its neighbors.

The number of events in one gait record is more than tens of thousands. If the construction is directly based on the entire event stream, there are over millions of edges in the graph, which consumes a large amount of time and memory to train the GCN. In order to improve the training efficiency and the generalisation of the model, some sampling methods are required, which need keeping the principal spatiotemporal features of the event stream.

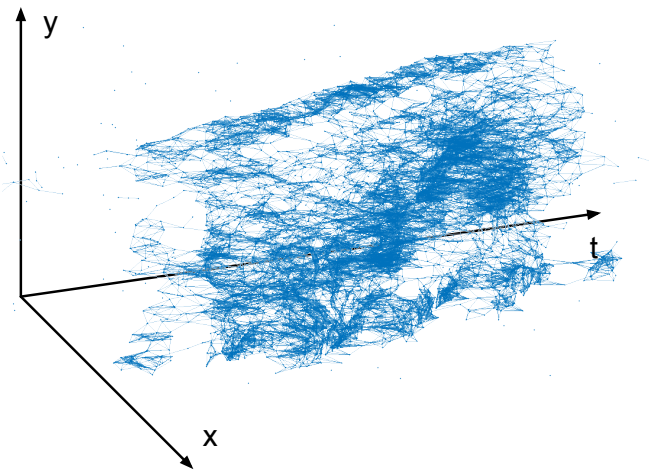


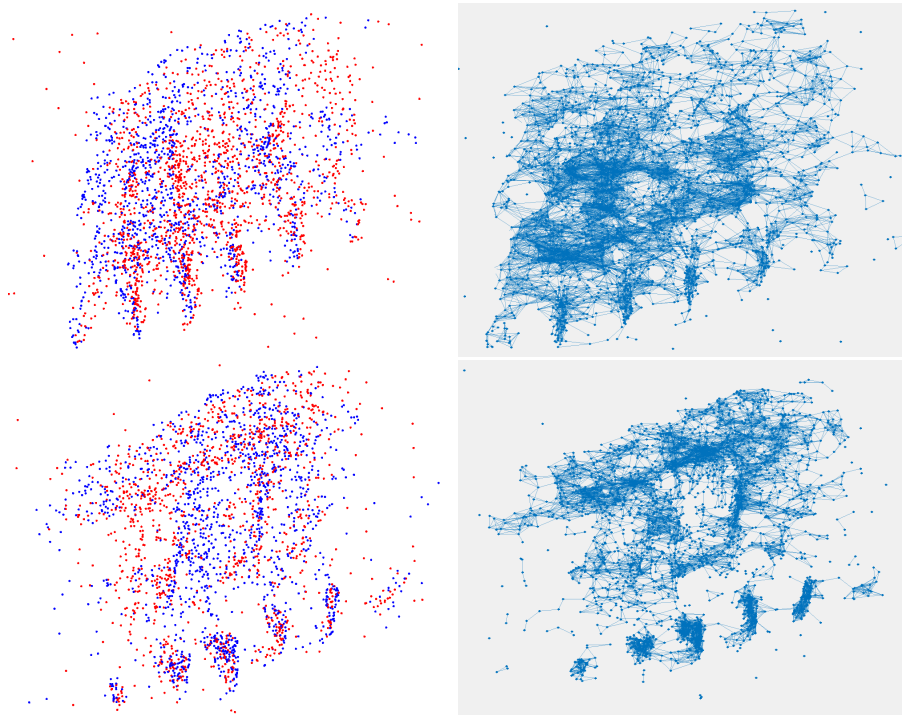
Figure 4.2: A graph-based representation of a event stream.

4.3 Event Sampling Strategy

The sampling strategy should keep the events' distribution in both spatial and temporal dimensions. Random sampling, a commonly used strategy, is employed to reduce the number of events and maintain the generalisation of the data. The octree is a data structure, which has been used to sample points according to the density, and OctreeGrid sampling strategy utilise this data structure's advantage to filter events depending on the spatiotemporal density of the stream.

4.3.1 Random Sampling

For random sampling, the events are filtered based on the percentage of the total number or a predefined number. Some examples of randomly sampled events are illustrated in Figure 4.3, where only 5% original events are left. Figure 4.3(a) shows the sampled events, and Figure 4.3(b) illustrates the graph after sampling. As can be seen from the figure, the constructed graphs still maintain strong connectivity, and the graph-based neural networks are applied to the sampled event stream. The random sampling strategy employed in the training phase can keep the distribution of the original data without losing the generalization ability.



(a) (b)

Figure 4.3: Randomly sampled events and the constructed graphs

4.3.2 OctreeGrid Sampling

OctreeGrid sampling [1, 115] can be used to reduce the number of events while the spatiotemporal structure of the original event-stream can still be well-preserved. The strategy firstly divides the whole 3D space into eight zones, and numbers them from zero to seven. The number of events in each zone is calculated, and if the number exceeds a predefined threshold, the corresponding zone will be further divided into eight small zones. Then, each zone is divided until that the number of events in this zone is equal to or less than the predefined threshold, and an event is randomly selected from the zone to represent this area. An example is illustrated in Figure 4.4(a). In the example, the number of events in the first zone exceeds the threshold. The first zone is then divided and number from 10 to 17. In the 11th zone, the number is still larger than the threshold. This zone is further divided into eight zones, from 110 to 117. An octree is employed to build the structure of these zones, which is shown in Figure 4.4(b). Each node with the number will randomly generate an event in its corresponding zone to represent this node. This strategy ensures that the density of

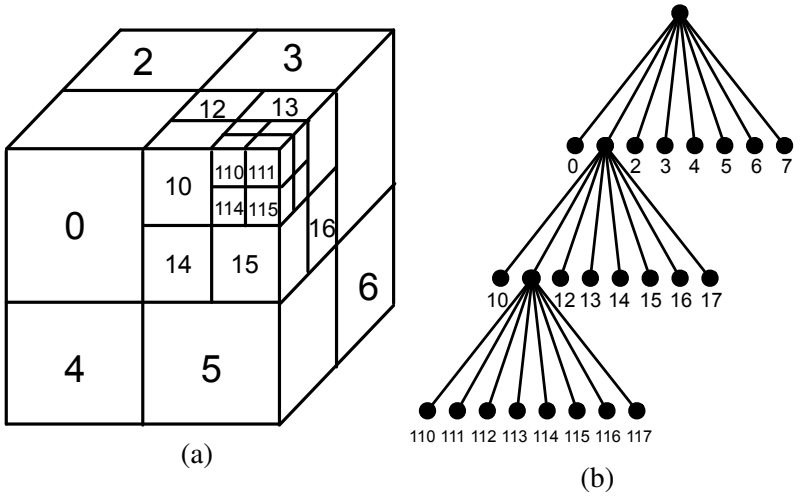


Figure 4.4: OctreeGrid structure (adapted from [115])

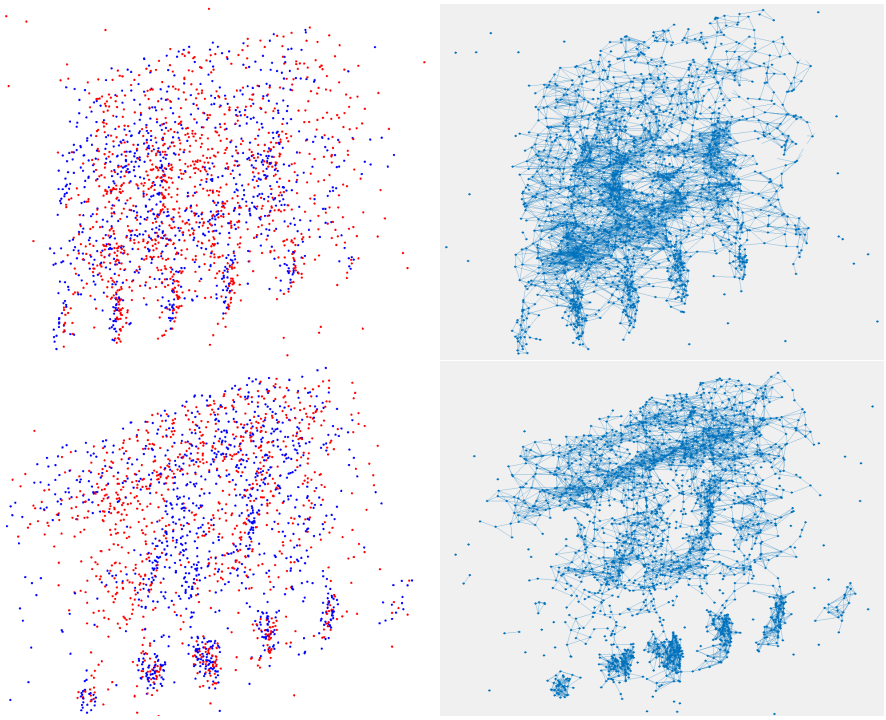
sampled events is similar to that of the original events, because the dense area will be divided smaller and more events will be left. The predefined threshold is the parameter to determine the maximum number of points in each leaf node (or grid) when building the structure of OctreeGrid, therefore, controls the downsampling rate.

The sampled events are shown in Figure 4.5 (a), and the corresponding graphs are illustrated in Figure 4.5 (b). Compared with the random sampling, there is no too dense area and the linked zone is larger. This structure is more similar to the original's, meanwhile decreasing the density of the points and edges.

4.4 Graph Neural Network Architecture

The workflow of Gait-3DGraph and the key components of the proposed GCN are shown in Figure 4.6. It starts with collecting event-streams consisting of hundreds of thousands events. Considering the computational complexity, the sampling strategy is applied to significantly reduce the number of events while preserving the spatiotemporal structure of the event-streams. The connectivity between the remaining events after downsampling is calculated according to the predefined radius of neighborhood to construct 3D-Graph representation of the event-streams. Finally, the 3D-Graphs are taken as the inputs to train GCN for event-based gait recognition.

After the event streams are downsampled and transformed to 3D-Graphs, we design a GCN-based deep recognition network for extracting features and recognising



(a) (b)

Figure 4.5: Sampled events using the OctreeGrid sampling and the constructed graphs

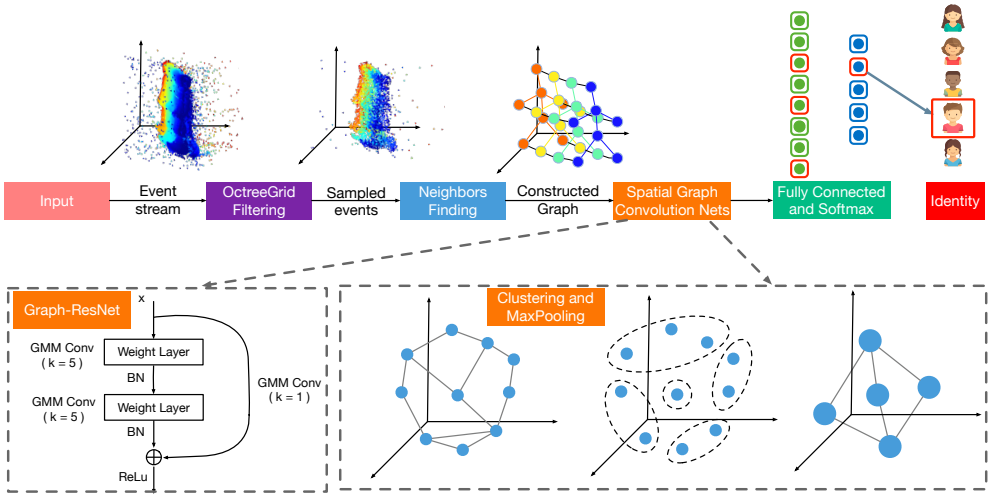


Figure 4.6: Workflow of 3DGraph-Gait.

human gaits. The key components of the network include Gaussian Mixture Model (GMM)-based graph convolution, Graph Residual Network, graph clustering and MaxPooling which are shown in lower part of Figure 4.6.

4.4.1 GMM-based Graph Convolution

Spatial-based convolution operation aggregates feature vectors among neighboring nodes by convolving with learned weights matrices to output a P-dimensional feature vector f'. The GMM-based convolution centered at node v_x can be expressed as weighted summation of J Gaussian kernels,

$$f'_p = \sum_{k=1}^{K} \sum_{y \in \mathcal{N}(x)} g_k w_k^p(u(x,y)) f(y) \quad p = 1, 2, 3, ..., P \tag{4.2}$$

where f'_p is one entry of the P-dimensional output feature vector. g_k is the weight associated to the k_{th} Gaussian kernel and $f(y)$ is the feature vector of node v_y. $\mathcal{N}(x)$ are the collection of the neighbors of the node v_x. The learnable weighting function $w_k^p(u(x,y))$ is defined on the pseudo-coordinates $u(x,y)$ for aggregating feature vectors of the neighboring nodes.

One of the key design factors of the graph-based convolutions is the choice of weighting functions or kernel functions such as B-spline kernels [65] and Gaussian Mixture Model (GMM)-based kernels [150].

In this chapter, we choose GMM-based kernel for convolution operations. Specifically, GMM-based convolution adopts K Gaussian models as the kernel functions and the weighting function of the k_{th} Gaussian model can be written as:

$$w_k(u) = \exp\left(-\frac{1}{2}\left(u - \boldsymbol{\mu}_k\right)^\top \boldsymbol{\Sigma}_k^{-1}\left(u - \boldsymbol{\mu}_k\right)\right) \tag{4.3}$$

where Σ_k^{-1} is the covariance matrix and the μ_k is the mean vector of the k_{th} Gaussian model. We denote the kernel size (number of Gaussian models) as K in the following manuscript.

The choice of the pseudo-coordinates is another important design factor for graph-based convolutions. In this chapter, we use relative Cartesian coordinates in three

dimensions (x, y, t) to estimate the relative position between neighbors so that both the spatial and temporal information can be extracted from the 3D-Graphs through GMM-based convolution.

4.4.2 Graph-ResNet Layer

The residual blocks have shown their ability in CNN-based neural networks [81]. They can improve the performance of the networks with the growth of the number of layers. The Graph-ResNet is similar to the residual blocks for the CNN, and the main difference is that the GCNcomponent replaces the CNN component. The Graph-ResNet layer of the GCN-based deep recognition network is designed according to the approach proposed in [27]. The major difference is the choice of the kernels and definition of the kernel size when operating graph convolution. Graph-ResNet is believed to be able to address the gradient degradation issue when the network depth goes deep. Lower-left of Figure 4.6 shows an example of the Graph-ResNet using GMM-based convolution. The kernel size K_1 in our Graph-ResNet is the number of Gaussian Models used for graph-based convolution (refer to Equation 4.3). Batch normalization (BN) is applied after each GMM-based convolution operation and a shortcut connection is added with kernel size $K_2 = 1$. As the results of our evaluation, the Graph-ResNet brings significant improvement on the recognition accuracy when incorporated in our GCN-based deep recognition network.

4.4.3 Graph Nodes Clustering and MaxPooling

Graph nodes clustering and MaxPooling strategy [194] is another important component in our approach. It is applied to reduce the complexity and alleviate the issue of over-fitting of when the network goes deep. MaxPooling aggregates feature vectors of the nodes in the same cluster to obtain the abstract representation so that the dense graph is transformed to a coarsen graph. The clusters are formed by evenly dividing the spatiotemporal space into 3D grids with size d (number of pixels) in each dimension, which is also known as pooling size. The nodes falling into the same grid will be merged together via MaxPooling. MaxPooling picks up the maximum value from dimension of the feature vectors of the nodes clustered together as the representation of

the corresponding node in the graph of the next layer. If the size of the spatiotemporal space in three dimensions is D_1, D_2, D_3 respectively, the maximum number of nodes after MaxPooling will be $\lfloor \frac{D_1}{d} \rfloor \times \lfloor \frac{D_2}{d} \rfloor \times \lfloor \frac{D_3}{d} \rfloor$.

4.4.4 Detailed Network Architecture

With the key components introduced above, we design a GCN-based deep recognition network for identifying gait from event-streams. The 3D-Graphs constructed from event-streams are taken as inputs to train the network. It starts with convolving the input graphs with a GMM-based Graph-ConvNet, GC_0(5,64), whose kernel size is 5 and output feature size is 64. A MaxPooling layer, MP_0(4), with grid size 4 is applied to merge the graph nodes from the first Graph-ConvNet layer. Then three Graph-ResNet layers, $GRes_1$(5,1,128), $GRes_1$(5,1,256) and $GRes_1$(5,1,512) with $K_1 = 5$ and $K_2 = 1$ are stacked sequentially whose output feature sizes are 128, 256 and 512 respectively. The resultant activations of ReLu [156] functions from each Graph-ResNet are passed to MaxPooling layers with pooling size $d = 6$, $d = 24$ and $d = 64$ respectively. At last, a fully-connected layer with 1024 nodes (FC(1024)) is connected to the last MaxPooling layer and softmax functions are used for obtaining the final recognition results. The detailed parameter settings of the network layers in sequence are GC_0(5, 64)- MP_0(4)- $GRes_1$(5, 1, 128)- MP_1(6)- $GRes_2$(5, 1, 256)- MP_2(24)- $GRes_3$(5, 1, 512)- MP_3(64)- FC(1024).

The proposed GCN processes the constructed graph of the event stream. In order to analyse the time and space complexity, we define V as the number of vertexes, E as the number of edges in the graph, K as the number of Gaussian kernels, and P as the number of the feature vectors to analyse the memory consumption and scalability. For the graph convolution layer, the time complexity is $O(KPE)$, and the space complexity is $O(PV + E)$. For the graph residual layer, the time and space complexity are the same as the graph convolution layer. Finally, the time complexity of clustering and pooling is $O(E)$, and the space complexity is $O(PV + E)$. The overall time and space complexities are therefore $O(KPE)$ and $O(PV + E)$. The number of original events is more than several tens of thousands, and the number of constructed edges is more than many tens of millions. It will consume a lot of time

and memory to process events in several seconds, and downsampling is definitely required. There are only about 2,000 events and several hundreds of thousands of edges after downsampling, which satisfies the real-time requirement.

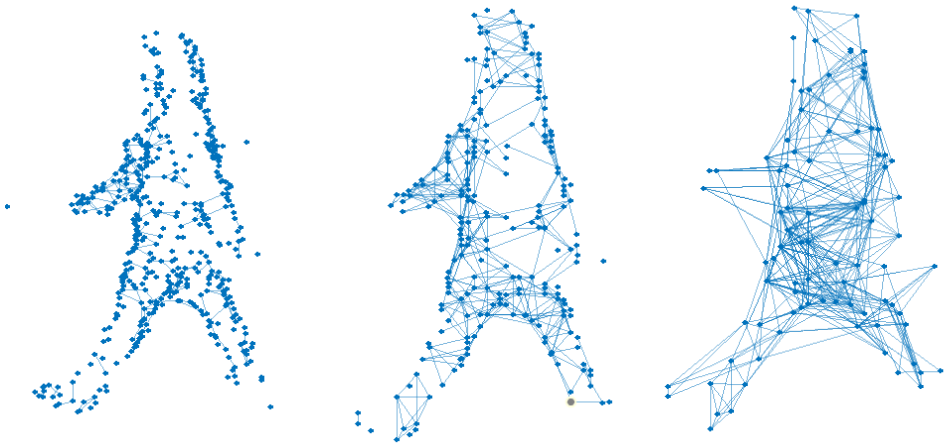
4.5 The Ensemble of 3DGraph-Gaits for Real-Time Recognition

Because the 3DGraph-Gait for gait recognition using a short partition of event streams cannot achieve the same accuracy using the full-length streams, some modifications are made for the graph-based representation to allow 3DGraph-Gait can be applied to a short stream that only contains several hundred events. One of the event cameras' advantages is their temporal resolution, which is several microseconds. In order to make use of this advantage, we choose the length of 100 events (about 8 milliseconds), 250 events (about 20 milliseconds), and 500 events (about 41 milliseconds) to build three base models and stack them as an ensemble network to achieve the competitive accuracy with the 3DGraph-Gait using the full-length streams.

4.5.1 Base Models

Three base models hold the same architecture described in Section 4.4, but are trained using different numbers of events. Unlike a full-size image that contains enough information, the length of an event stream decides the useful volume of information for gait recognition. Insufficient information leads to the network's inability to recognise people, while superfluous information increases the difficulty of features extraction, losing accuracy. Meanwhile, the difficulty of gait recognition also varies depending on not only the length but also the walking status of the target. In other words, the long event stream does not always perform better than the short event stream. Therefore, the three base models are trained for different stream lengths and the stack as an ensemble network for gait recognition with short-time event streams.

Compared with several hundred thousand events of the full-length stream, 500 events, 250 events and 100 events are relatively small. 3DGraph-Gait is able to process these events directly, so no sampling strategy is adopted in all base models. As the



(a) graph with 500 events (a) graph with 250 events (a) graph with 100 events

Figure 4.7: Constructed graphs for different base models

scale of the time domain is not adaptive to the space domain, the normalisation is conducted on the time domain and can be computed as:

$$t_i' = \frac{t_i - t_1}{t_n - t_1} \times \min(X, Y) \tag{4.4}$$

where t_i' is the normalised timestamp, t_i is the original timestamp of the ith event, X and Y are the height and width of the receptive field of the event camera. This normalisation aligns the scale of the time domain to that of the space domain. For graph construction, the parameters in Equation 4.1, α and R, are set as 1 and 5, respectively, which are same as the original 3DGraph-Gait.

Streams with 250 events and 100 events are more sparse than the streams with 500 events after mapping into the same size space. After mapping these events using the Equation 4.4, a larger radius should be used to construct the graph, and R in Equation 4.1 for 250 events is set as 10, and that for 100 events increases to 20. The constructed graphs are illustrated in Figure 4.7. The model for a large number of events will focus more on details and local features, while that for a small number of events makes the most of the whole structure. Each base model extracts different timescale features from the event stream, and the ensemble of these base models can achieve better accuracy than each of them.

4.5.2 Attention-Based Ensemble Method

When the base models are ready, the ensemble approach directly decides the accuracy of the final predictions. Because the structure of each base model is nearly identical, we use the logits, the features before the softmax layer, as the input for the ensemble rather than the low-level features and final probabilities. Although different base models extract different time scale features, there are some similar features, so directly fusing these features incurs redundancy. In contrast, the final prediction is not enough to estimate the confidence or relationship between base models. Therefore, the logits as the middle features are selected for fusion.

This attention-based ensemble method is visualised in Figure 4.8. Similar to the attention mechanism, the logits of base models are utilised to generate the corresponding weights. The same fully connected layer with the same parameters is employed for the base models, following the generation of logits. The target of this layer is to compute the confidence of base models for gait recognition. If all base models have identical predictions, the final prediction is with high probability, but if the predictions are different, a decision-making strategy should solve this problem. Because each base model is trained separately, no attention or confidence is included. The additional attention method is to generate the attention for outputs according to the logits, which can be donated as:

$$att_i = F_{att}(logits_i) \tag{4.5}$$

where F_{att} is the shared attention layer and $logits_i$ is a feature vector.

Finally, the logits are aggregated according to the generated attention, and the softmax is utilised to compute each class's probability, which can be calculated as:

$$prediction = \arg\max(softmax(\sum_{i=1}^{n}(att_i \times logits_i))) \tag{4.6}$$

where $logits_i$ is the logits of the ith base model, att_i is the corresponding attention, and n indicates the number of base models. The final result has the same shape with $logits_i$ and the prediction is the class with the highest probability.

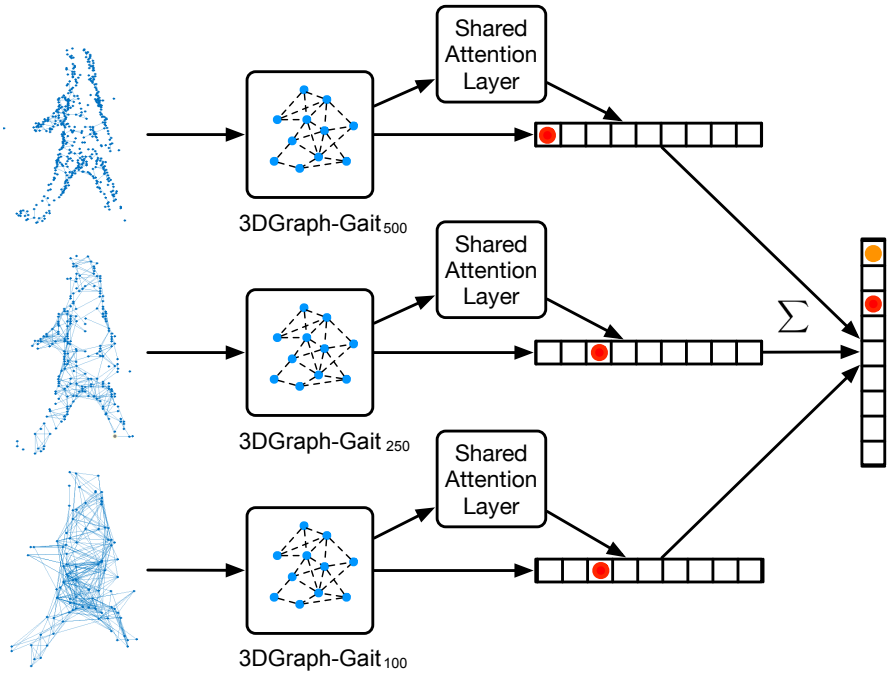


Figure 4.8: The ensemble network of 3DGraph-Gaits

4.6 Evaluation

Firstly, in order to determine appropriate parameter settings of 3DGraph-Gait, we evaluate the recognition accuracy of the proposed method by varying settings, including sampling strategies, MaxNumEvents, neighboring range, last pooling size, convolution kernel size, the influence of Graph-ResNet and complexity of the network architecture. Secondly, 3DGraph-Gait is compared with other event-based gait recognition approaches in overall accuracy, the number of training samples and resource consumption. Finally, the ensemble network of 3DGraph-Gait is evaluated for real-time recognition accuracy.

4.6.1 Evaluation on Sampling Strategies

Sampling Method	Random Sampling	OctreeGrid Sampling
Accuracy	91.0±1.1%	93.0±0.8%

Table 4.1: Recognition accuracy using different sampling strategies

As GCN cannot directly deal with the entire event stream, sampling strategies act as a pre-processing component to remain a proper number of events, and the similarity

between the remaining events and the original events affects the accuracy of GCN. In this part, we evaluate the effects of different sampling strategies on the accuracy, and the results are shown in Table 4.1. OctreeGrid sampling strategy can stably keep the remaining events with a similar spatiotemporal density with the originals, which achieves about 93% accuracy and has a lower deviation. The random strategy cannot strictly generate similar distribution, and thus, the accuracy is lower than OctreeGrid, about 91.0%, and the deviation is also large. In summary, a dedicated sampling strategy, OctreeGrid, outperforms the random sampling.

4.6.2 Comparison with Different Event-based Gait Recognition Approaches

After quantitatively evaluating the effects of various parameters and components of 3DGraph-Gait on the accuracy, we compare the recognition accuracy of different event-based deep recognition networks. We also include SVM as a benchmark to determine if deep neural networks are necessary for the event-based gait recognition task. Besides 3DGraph-Gait and EV-Gait, the other competing approaches are as follows:

2DGraph-3DCNN [28] was proposed to extract the spatiotemporal features from event streams. It splits each event stream into multiple slices over time. For each slice, a very short-term of period (e.g., 30ms) is picked to construct a 2D-Graph and spatial features are extracted through graph-based convolution with a B-spline kernel [65]. Then the 2D-Graphs are transferred to a grid representation through Graph2Grid operations [28]. Finally, 3DCNN [249] is applied to extract the spatiotemporal features for action recognition. We optimize the network structure of 2DGraph-3DCNN for the gait recognition task, and the final detailed network settings are GC_0(5, 64) - MP_0(2) - GC_1(5, 128) - MP_1(4) - Graph2Grid(8, 32, 128) - $3DConv_0$(3, 128) - $3DMP_0$(2) - $3DConv_1$(3, 256) - $3DMP_1$(2) - $3DConv_2$(3, 512) - $3DMP_2$(2) - $3DConv_3$(3, 512) - $3DMP_3$(2) - GA(512) - FC(256) - Dropout(0.5). GC(5, 64) is a graph-based convolution with kernel size 5 and output feature size 64. MP(2) and 3DMP(2) are two or three dimensional MaxPooling layer with pooling size 2 for each dimension. Graph2Grid(8, 32, 128) converts a stack of graphs constructed from 8 slices of event-stream to eight 32×32 matrices and the output feature size

(depth) is 128. 3DConv(3, 512) is 3D-convolution layer with kernel size 3 and output feature size 512. GA(512) merges multiple features from previous layer into a one-dimensional global feature. Finally, FC(256) is fully-connected layer with 256 nodes and Dropout(0.5) randomly throws half of the coefficients to alleviate the problem of overfitting.

LSTM-CNN is based on EV-Gait but considers the variance of human gait through time. It splits the whole event-stream into multiple slices and we set the number of slices as 8 which produces the highest accuracy. Each short-term slice is converted to the image-like representation. The CNN-based network inheriting from EV-Gait is applied to extract spatial feature from each slice. Then the sequence of feature vectors are taken as the input of a LSTM network with 100 hidden states to recognize gaits from the event-streams.

SVM-PCA is a benchmark method to determine if the deep neural networks are necessary for our event-based gait recognition task. SVM-PCA adopts the same event images as EV-Gait and concatenates the event images by columns to form high-dimensional vectors. Principal Component Analysis (PCA) is applied on the high-dimensional vectors to extract features (we set the output dimension of PCA as 500) to train SVM-based classifier for gait recognition.

Comparison on Best Accuracy. We compute the average and standard deviation of the recognition accuracy of the five competing approaches over 30 independent training and inference trials. The parameters of the five approaches are all carefully tuned and the best averaged accuracy is reported in Table 4.2. The results show that, the two approaches with graph-based representations achieves significantly higher recognition accuracy than those with image-like representations and the gap is up to 8.4% (94.9% v.s. 86.5%). By further comparing the two graph-based approaches, we find the 3D-Graph representation produces higher recognition accuracy than 2D-Graph as it can better preserve the spatiotemporal information of the asynchronous event streams than a sequence of discrete 2D-Graphs in a human gait recognition task. It is worth noting that, the accuracy of SVM-based classifier cannot compete with the four deep learning approaches and the difference is up to 16.5%.

There are a number of reasons that 3D-Graph representation generates the highest

Methods	3DGraph-Gait	2DGraph-3DCNN	EV-Gait	LSTM-CNN	SVM-PCA
Accuracy	94.9±1.5%	92.2±2.1%	87.3±0.9%	86.5±0.8%	78.05%

Table 4.2: Recognition accuracy of different event-based gait recognition approaches

recognition accuracy among the competing approaches for event-based gait recognition. First, image-like representations suffer from misalignment and noisy background issues, the recognition accuracy cannot be guaranteed if the distance between the walking subject and the camera is not well-controlled, which leads to various scales of the recorded subject. However, event stream alignment is challenging and still remains unsolved. On the contrary, graph-based representation focuses on the moving subject in the view directly, therefore alleviates the influence of misalignment and background noises. 2DGraph-3DCNN employs 2D-Graphs for spatial feature extraction, however, the 3DCNN component requires careful alignment when mapping the 2D-Graphs to grid representation. Finally, 2DGraph-3DCNN converts the event stream to discrete 2D-Graphs by picking up very short-term period from the slices of the event stream and the information in between is discarded. While 3DGraph-Gait takes the whole event-stream as an entirety and preserves most of the shape when constructing the 3D-Graph.

Comparison on Number of Training Samples. We then compare the recognition accuracy of the event-based approaches with respect of the amount of training samples per subject. The amount of samples per subject required for training is important as few shot learning can save significant training efforts especially when registering new subjects. It has significant impact on user experience. In particular, we randomly select different number of training samples from each subject, varying from 5 to 100. For each case, we retrain all the four event-based approaches for 30 times and report the average and standard deviation of recognition accuracy. Table 4.3 shows the results, and we see that as more samples are used in training, the recognition accuracy of all approaches grows, but with different growth rates. By comparing the results across different approaches, EV-Gait produces significantly higher recognition accuracy than graph-based approaches when number of training samples is low and becomes almost level after 10 or more training samples are used. The accuracy of

3DGraph-Gait surpasses other approaches when sufficient training samples (over 50 in the table) are used. This indicates that EV-Gait doesn't require massive training data to converge so the image-like representation is the choice when only limited number of training samples are available. In contrast, 3DGraph-Gait shows a significantly higher performance cap than those using image-like representation, therefore, is more preferable when sufficient training data could be sourced.

Samples\Methods	3DGraph-Gait	2DGraph-3DCNN	EV-Gait	LSTM-CNN
5 Samples	36.3±2.2%	6.1±1.3%	**79.9**±2.0%	33.1±6.3%
10 Samples	61.7±4.1%	14.5±2.7%	**85.9**±1.8%	57.6±8.6%
20 Samples	78.6±1.1%	48.2±7.3%	**86.5**±0.7%	76.0±4.2%
50 Samples	**90.0**±1.2%	82.9±3.2%	87.2±0.8%	85.5%±1.4
100 Samples	**94.9**±1.5%	92.2±2.1%	87.3±0.9%	86.5±0.8%

Table 4.3: The recognition accuracy of event-based deep recognition networks with different number of training samples per subject

Comparison of Resource Consumption. In addition to the recognition accuracy, the resources consumption of the event-based gait recognition approaches are also important for practical use. We implement both 3DGraph-Gait and EV-Gait on Intel UP Board [4] with a Quad-core 1.44Ghz Intel Atom x5-Z8350 microprocessor on board. The RAM of the board is 1G and ROM is 16G. The operating system is Ubuntu 16.04. After implementation, we profile the resources consumption of the total number of coefficients, averaged inference time, memory usage and energy consumption of the proposed event-based gait recognition approaches. The number of coefficients can be conveniently obtained from Pytorch API. Average inference time and memory usage can be drawn from the system when running the programs. We use external tool to monitor the power consumption (current and voltage) of the board when running the inference of different event-based gait recognition approaches.

Methods	3DGraph-Gait	EV-Gait
Number of Coefficients	7.15 M	64.61 M
Average Inference Time	436.23 ms	238.43 ms
Memory Usage	410.87 Mbytes	413.05 Mbytes
Energy Consumption	0.876 J	0.238 J

Table 4.4: Resources consumption of 3DGraph-Gait and EV-Gait on UP board

The resources consumption of gait recognition on UP board are shown in Table 4.4.

First of all, the average inference time is acceptable on the resource-constrained platform; the inference can made within half second with the slower approach (3DGraph-Gait). Then by comparing the resources consumption of different EV-Gait approaches, we can observe the number of coefficients of GCN-based approach is only about one ninth of CNN-based approach (7.15 million v.s. 64.61 million), however, it requires almost the same running memory (410.87Mbytes v.s. 413.05Mbytes), 1.8 times inference time and 3.7 times energy consumption compared with CNN-based approach. Our conjecture is because the graph-based convolution is based on an extension library for Pytorch (Pytorch Geometric [2]) which is implemented by third-party and not well optimized. Therefore, we can claim that, with the popularity of GCNs, the resources consumption of the GCN-based approach can be significantly reduced when proper optimization on the implementation of graph convolution are available in the future.

Comparison with the other geometry-based approach. As an event can be expressed as (t, x, y, p), the projection of a stream of events' timestamps and locations into a 3D space resembles a point cloud. The main difference is that the space of the point cloud only describes spacial information, whereas the space of the projected events includes spatial and temporal information. To leverage such similar features of the point cloud, a Point-Net-based neural network [226] is designed for gesture recognition of event cameras. Inspired by the power of graph neural networks for point clouds, we introduce this kind of neural network for events. Compared with PointNet, the graph neural network can extract and propagate features along the edge, which may achieve a higher accuracy of gait recognition with event cameras. Here, we chose the PointNet as a baseline to verify the advantage of graph neural network. The result is presented in Table 4.5:

Sampling Method	PointNet	3DGraph-Gait
Farthest Point Sampling	50.5±1.8%	86.9±2.3%
Random Sampling	64.5±1.3%	91.0±1.1%
OctreeGrid Sampling	67.2±0.9%	93.0±0.8%

Table 4.5: Recognition accuracy using geometry-based different neural networks with different sampling strategy

As can be seen from the table, the graph neural network, 3DGraph-Gait, out-

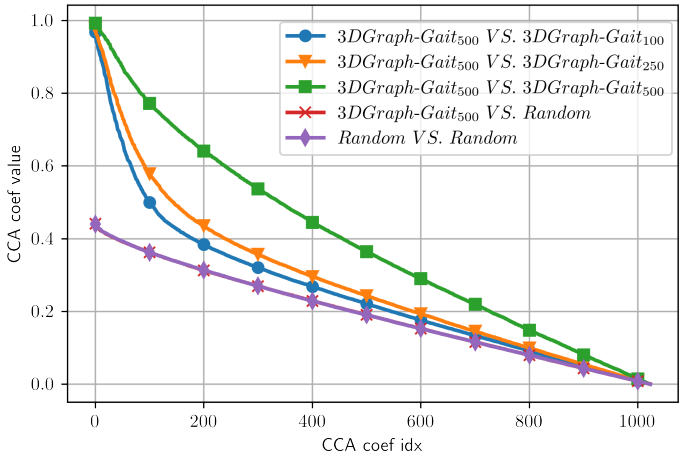
performs PointNet, which is designed for point clouds. Compared with the shape and the local features extracted from the PointNet, the rich features extracted from 3DGraph-Gait are more relevant to the relationships and links between events, which is more critical for gait recognition tasks.

4.6.3 Evaluation of the Ensemble Network for Real-Time Gait Recognition

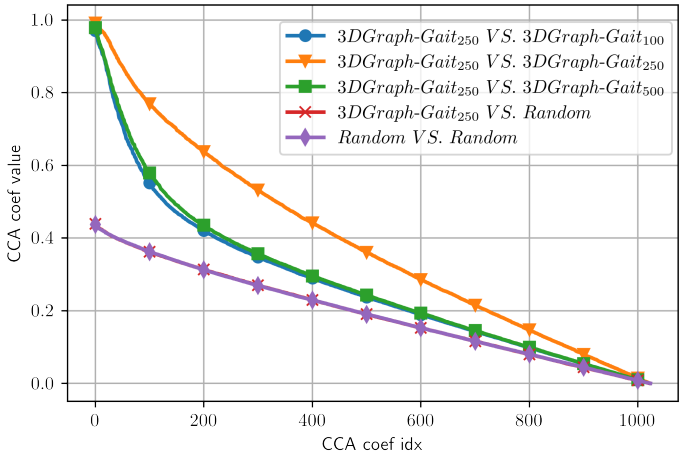
Previous experiments are based on the entire event streams for gait recognition, but these streams are not suitable for investigating real-time gait recognition performance. Therefore, we divide an entire stream into several sequential short event streams, and each of them includes continuous 500 events. The following evaluations are conducted using these short streams.

Firstly, the similarity between the features extracted from the base models is analysed. Because the ensemble utilises the features from the base models, the accuracy increase should be limited if these features are similar. Canonical correlation analysis (CCA) is a tool to evaluate the similarity of the features from different neural networks and further analyse the similarity of these networks. It finds bases for two feature matrices, so the correlation is maximised when the original matrices are projected onto the bases [105]. Here, the features before the fully-connected layers are acquired from the will-trained base models for comparison. Other randomly generated matrices are employed as the baseline. The results are shown in Figure 4.9. As can be seen, the difference between 3DGraph-Gait$_{500}$ and 3DGraph-Gait$_{100}$ is larger than the difference between 3DGraph-Gait$_{500}$ and 3DGraph-Gait$_{250}$. With the different numbers of used events increasing, the extracted features are more different. The gap between the features from the same models and that from different models shows that these features from the different base models are different enough for the ensemble.

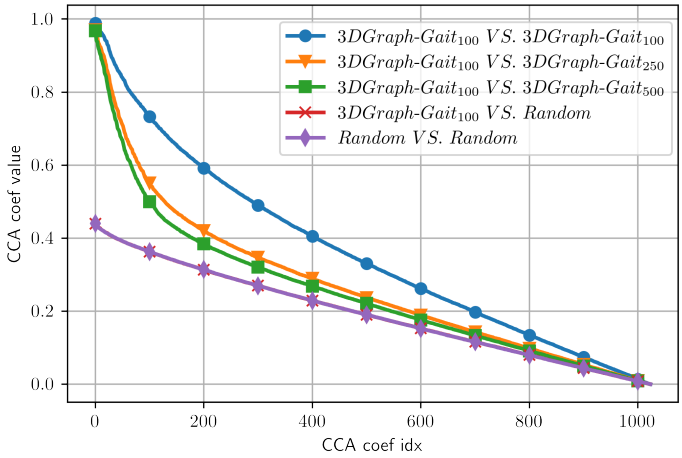
Secondly, we evaluate the overall accuracy, and the results are shown in Table 4.6. Due to the radius limitation, the accuracy of 3DGraph-Gait for 500 events is near 90.0%, which is relatively lower than other base models. In contrast, although the radius of 3DGraph-Gait for 100 events is large enough, the accuracy is about 90.4% because of the limited number of events. The 3DGraph-Gait for 250 events, benefiting



(a) 3DGraph-Gait$_{500}$



(b) 3DGraph-Gait$_{250}$



(c) 3DGraph-Gait$_{100}$

Figure 4.9: CCA similarity between the features extracted from different base models

from the balance of events and the radius, achieves the best performance in base models. Compared with these based models, the ensemble network achieves the highest accuracy, which means that multi-time scale features describe the different patterns of gait recognition and the ensemble network can make the most of them. Therefore, the ensemble network is better than base models for real-time gait recognition.

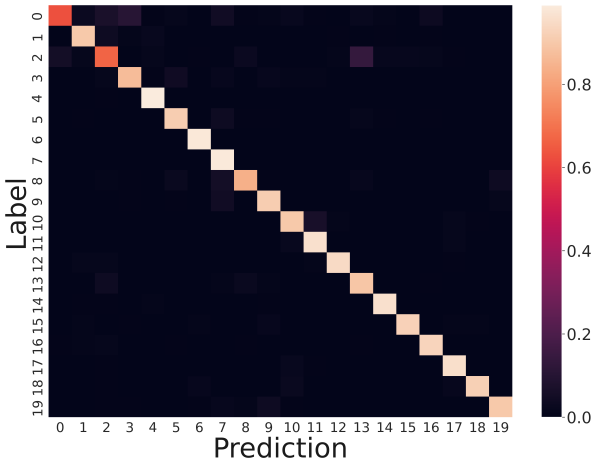
Model	3DGraph-Gait$_{500}$	3DGraph-Gait$_{250}$	3DGraph-Gait$_{100}$	Ensemble
Accuracy	89.7±0.4%	93.0±0.3%	90.4±0.2%	94.6±0.4%

Table 4.6: Accuracy of base and ensemble models for real-time gait recognition

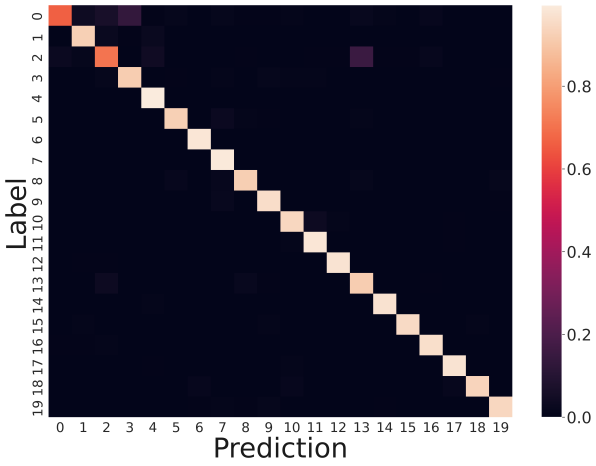
The ensemble mechanism benefits from the diverse accuracy for the different classes of each base model. Thus, we then further evaluate the accuracy for the different classes using confusion matrixes. The confusion matrixes of based models are visualised in Figure 4.10, while that of the ensemble network is illustrated in Figure 4.11. Although there are some similarities between base models, the accuracy for each class still has some differences. For example, the accuracy for class 0 of 3DGraph-Gait$_{100}$ is worse than the other base models, but that for class 17 is better than others. The ensemble network achieves the best accuracy for each class compared with all base models, which implies the ensemble method can effectively fuse the generated logits of each base model and make the decision.

Finally, we evaluate the accuracy varying over time from the beginning of the event streams, and the results are shown in Figure 4.12. At the beginning, the target has not entirely entered the surveillance area, so the accuracy is relatively low. When the entire gait pattern is fully captured, i.e., after 5000 events passing, the accuracy of base models for 250 events and 500 events increases to more than 90%, and that for 100 events is more than 80%. The ensemble network consistently achieves better accuracy than these base models, demonstrating its ability to apply in real-time gait recognition scenarios.

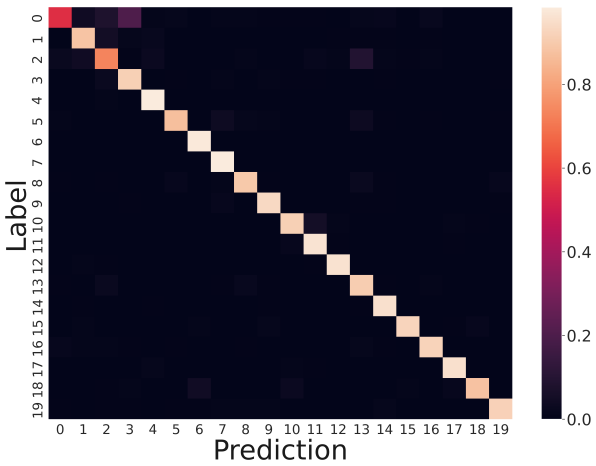
To sum up, extending 3DGraph-Gait for real-time gait recognition achieves the competing accuracy with the recognition using the entire event stream, which reveals the possibility for real-time gait recognition using GCN-based architecture. Multi-time scale features can improve gait recognition performance compared with single time



(a) 3DGraph-Gait$_{500}$



(b) 3DGraph-Gait$_{250}$



(c) 3DGraph-Gait$_{100}$

Figure 4.10: Confusion matrixes of base models

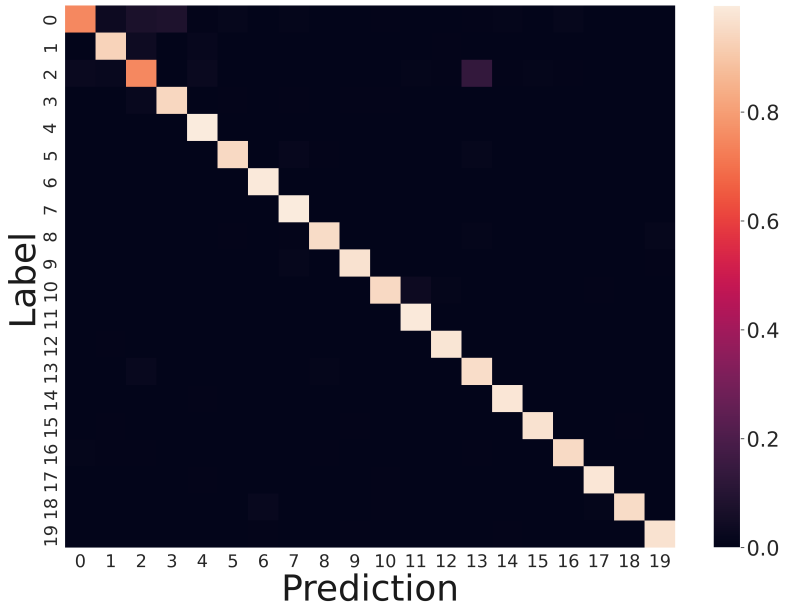


Figure 4.11: Confusion matrixes of the ensemble of 3DGraph-Gaits

scale features, and the attention-based ensemble method can effectively aggerate the logits from based models. The ensemble network outperforms each based model for both overall accuracy and time-varied real-time accuracy.

4.7 Summary

In this chapter, we have proposed another representation approach for event streams. Compared with the previous image-like representation, this graph-based representation focuses on the inherent spatiotemporal relationship between events and builds a three-dimensional structure rather than a two-dimensional image. In order to extract spatiotemporal features from the graph, a GCN-based approach, 3DGraph-Gait, is designed for gait recognition, which performs graph convolution among sampled events to generate features and performs the classification. For real-time gait recognition with fewer events, 3DGraph-Gait is extended to support short event streams, and the ensemble of multi-time scale 3DGraph-Gaits can achieve nearly the same accuracy with the entire event stream. Some lessons can be learnt from this chapter:

- The graph-based representation can maintain spatiotemporal features, which

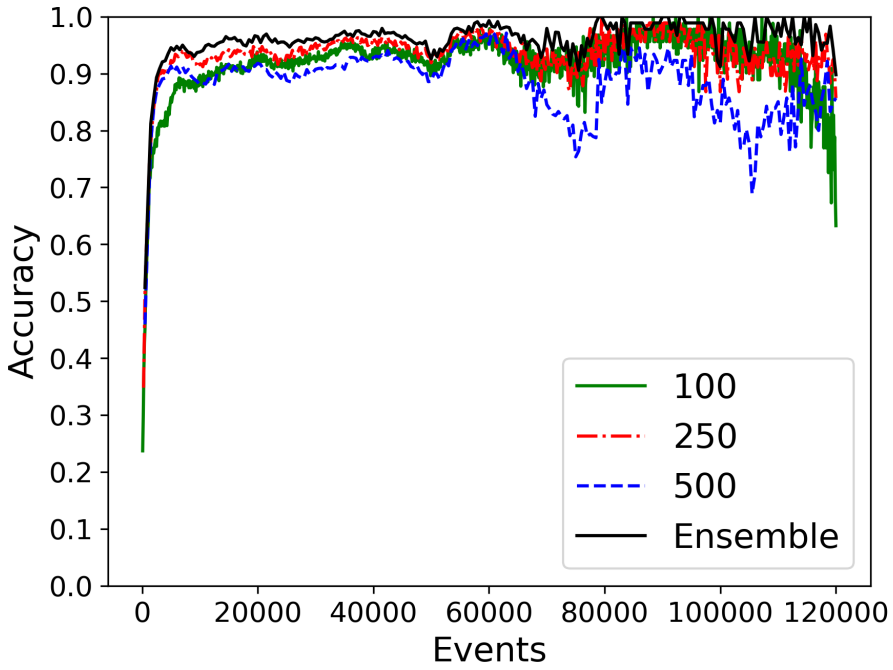


Figure 4.12: Gait recognition accuracy at different place of event streams

is more effective for gait recognition than the image-like representation. The image-like representation reduces the time dimension, although the time surface preserves the time information of the last events in each pixel. This representation well keeps the visual information but ignores the temporal features. In contrast, the graph-based presentation can solve this problem, which constructs a 3D graph to present even streams. The drawback of graph-based representation is that an entire event stream cannot be all constructed due to a large number of events, and the sampling strategy should be applied.

- Sampling strategies affect the performance of the GCN-based approaches for gait recognition. Because of a large number of events, the graph construction cannot use all events, and sampling strategies directly affect the final performance. The strategy should keep the same distribution and density in global space and each dimension, and the Octree Grid sampling strategy outperforms the random sampling strategy for gait recognition.

- 3DGraph-Gait, a GCN-based approach, can effectively extract gait-related

spatiotemporal features from the constructed graph, including GMM-based graph convolution layers, Graph-ResNet layers, graph node clustering layers and max-pooling layers. This GCN-based approach outperforms EV-Gait for gait recognition.

- The original 3DGraph-Gait is designed for the entire event stream, and furthermore, can also extract some features based on a few events after extending the graph-based representation. The ensemble of several 3DGraph-Gait base models based on different time scales can boost the performance of real-time gait recognition with a short part of the event stream.

In summary, we proposed 3DGraph-Gait, a graph-based representation and learning approach for gait recognition with event cameras in this chapter. This approach can effectively extract spatiotemporal features from both the entire event streams and a short part of them. The ensemble of multi-time scale 3DGraph-Gaits can achieve about 94% accuracy with only 500 events (accumulated in approximately 5 ms).

Some related challenges should be further explored:

- When event cameras keep monitoring some areas, will some information unrelated to gait recognition be inferred?

- Is there a mechanism compatible with the generation rate and transmission requirement for event cameras that also ensures security?

Chapter 5

EV-Encryp: Efficient Encryption Framework for Gait Recognition with Event Cameras

5.1 Introduction

Event cameras have demonstrated their capability in gait recognition, activity recognition [173], aided driving [146], localisation [245], and anomaly detection [42]. With event cameras applied for recognition and other privacy-related tasks, security is another major concern that should be resolved. Gait recognition, a sensitive task, may expose personal position information and access statues to some areas and even be copied for authentication and other security-critical tasks. However, no security mechanism has been designed for event cameras. In this chapter, we investigate the security risks and privacy challenges of event cameras, and propose an encryption framework to protect gait recognition with event cameras and other general tasks.

It is certain that unprotected images and videos would threaten users' privacy and security, and accordingly, there exist many mature approaches (such as [76, 88, 169, 174, 231, 235]) in solving traditional image and video security issues. Unlike RGB cameras which produce pixel-based images, an event camera outputs a stream of events, which captures luminous intensity changes at each pixel in a fine-grained

manner. Prior work assumed that event cameras are secure and privacy-preserving simply because it does not produce visible images, and directly applied event cameras in privacy-related scenarios. For example, Samsung developed an event camera-based in-home monitoring solution named *SmartThings Vision* [183]. However, the occurrence of grayscale image reconstruction approaches [177, 223, 242] seriously threatens the security of event cameras. These approaches can be used to generate visible images from the event cameras' output, and therefore all security issues related to traditional images and videos remain existing and applicable to event cameras' applications, which are worth revisiting and investigating.

To enhance the security of events of event cameras with affordable overhead in communication and storage, we analyze potential security risks, design a threat model dedicated for event cameras' data, and propose a novel encryption framework that can protect event cameras' data against image reconstruction and human identification attacks with efficient performance on various platforms. Major contributions in this chapter include:

- Based on existing datasets, algorithms and systems, we have proposed a novel privacy threat model dedicated for events. The adversary's target is to perform gait recognition and even reconstruct visible grayscale images from an event stream.
- We have proposed and designed an efficient encryption framework for events, which employs 2D chaotic mapping and effectively protects events against human identification and grayscale image reconstruction.
- Extensive experiments have demonstrated both effectiveness and high efficiency of the proposed framework on a wide range of platforms, including resource-constrained devices.

The rest of this chapter is organized as follows. In Section 5.2, we analyze security issues and privacy challenges related to event cameras, and propose a dedicated threat model. In Section 5.3, we propose a novel encryption framework with in-depth elaborations. In Section 5.4, we evaluate the framework in terms of its security and

efficiency, using different datasets on various devices. Finally, we conclude this work in Section 5.5.

5.2 Security Risks and Privacy Challenges

5.2.1 Event-based Applications in Private Scenarios

Taking advantage of event cameras, some datasets, approaches and solutions have been proposed in privacy-related scenarios. Samsung designs SmartThings Vision [183], an event camera-based home monitoring vision sensor. It can detect unexpected intruders and alert fall by analysing the movements with a privacy-preserving approach. This is the first event camera-based commercial product which employs its characteristics in security domains.

Face and eye tracking are also sensitive tasks, which involve facial and iris information. A direct face detection approach has been designed in [17], which utilises random forest and chooses the histogram of oriented gradients as features. Besides, an eye-blink tracking algorithm combined with face detection has been proposed in [117]. It firstly captures the temporal signature of an eye blink and then detects the face by the recognized eyes. In [175], kernelized correlation filters have been employed for face detection, and an event-based face dataset was published. Furthermore, an event-based near-eye gaze tracking dataset was made available to the public. These algorithms and datasets extended the applications of event cameras in private scenarios, but the security issues in these scenarios deserve further attentions, since the event cameras' outputs can be reconstructed to traditional images.

Besides facial and iris features, other biometrics can also be used for human identification. In [196], a human identification approach is directly drawn from the gait recognition in RGB videos. This approach involves five phases: visualization of the event stream, human figure detection, estimation of optical flow, human pose estimation, and gait recognition based on neural features. DVS128-Gait has collected several gait datasets using event cameras and trains a CNN to tackle the gait recognition problem. 3DGraph-Gait has applied a GCN to identify human by the gait. These algorithms can even utilise event cameras' outputs for recognition without the

reconstruction of images or videos. As a countermeasure, encryption can effectively prevent recognizing objects and other unauthorised access, to protect the outputs in more general scenarios.

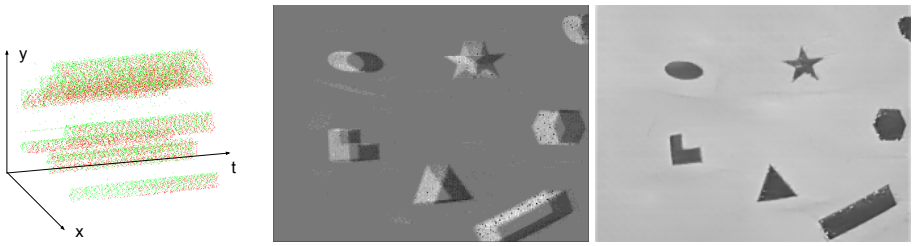
5.2.2 Security Risks of Event Cameras

Event cameras have shown their capability to solve some privacy-related tasks. However, some algorithms can be used to reconstruct high-resolution grayscale images from the stream of events, and it threatens the privacy of event camera-based applications. Given the requirement to apply existing vision algorithms to event cameras, reconstruction has been accompanying event cameras since its appearance. In [17], a patch-based dictionary is learnt from event streams offline, and reconstruction is executed online based on the dictionary. A variational model is proposed to estimate the behaviour of a event camera, and the grayscale images are reconstructed from the model [155]. A self-supervised learning approach is proposed in [167], which combines optical flow and event-based photometric constancy.

In addition to grayscale image reconstruction, some video synthesis algorithms have been further proposed. E2VID [177] is an end-to-end neural network-based approach, which is trained with the data generated from the simulator. This approach shows a good generalization with real-world data. Generative Adversarial Networks (GANs) have been used to generate videos from event streams. Both conditional GAN [223] and enhanced Cycle-GAN [242] have shown their capabilities in generating high-quality videos.

Some works managed to generate super-resolution intensity images from events. EventSR [224] utilizes three neural networks to complete reconstruction, restoration and super-resolution tasks, respectively. Besides, another end-to-end neural network for super-resolution reconstruction is proposed in [46], which pairs the events and the optical flow to generate images.

These reconstruction and generation approaches have brought traditional vision algorithms into the domain of event cameras, together with the security issues associated with traditional cameras. These approaches are treated as a means of attack for acquiring privacy-related visual information.



(a) (b) (c)

Figure 5.1: The different approaches to visualise a sample event stream

Figure 5.2: Reconstructed images from the DAVIS dataset [154] and DDD17 [29]

5.2.3 Privacy Challenges and Threat Model

Compared with a traditional RGB-based image, an event frame lacks detailed visual information, which drives its application in private scenarios such as in-home surveillance. Fig. 5.1 (a) illustrates a sample event stream in the duration of 0.1 second, while red and green points represent positive and negative events, respectively. However, some new algorithms can reconstruct grayscale frames from a series of events, threatening privacy-related applications. For example, events in a short period can be directly accumulated and normalized as an image as Fig. 5.1 (b), and a reconstructed image is shown in Fig. 5.1(c) using the approach in [177], whose angle, edge and other details can be clearly recognized. More real-world reconstructed images are illustrated

Figure 5.3: Failed reconstruction images based on the DVS128-Gait dataset [227]

in Fig. 5.2. These reconstruction algorithms are treated as a means of visualisation attacks. However, such attack may not be successful at all times. When the number of events is limited or no event appears in an area, the reconstructed images are blurry and cannot be recognised clearly. Some examples of failed visualization attacks are illustrated in Fig. 5.3. However, this does not imply that attackers cannot obtain any valuable information. Event-based recognition algorithms commonly enable attackers to acquire desired information without image reconstruction. One example is a variety of human identification algorithms, which reaches up to 90% accuracy. Due to the privacy issues in identification tasks, these identification approaches are treated as another kind of attack, namely recognition attack. Based on aforementioned assumptions, the objectives, capabilities and knowledge of the threat model are defined as follows:

Objectives. The adversary wants to (i) reconstruct the visible images from the event streams (visualization attack) and/or (ii) identify different people (recognition attack) under the condition that no clear grayscale image can be reconstructed.

Capabilities. The adversary can access events during the transmission and storage processes. The adversary may not acquire the continuous integrated event stream, but a part of (maybe nonadjacent) events for attacking.

Knowledge. The adversary is supposed to know the encoding format of events and is able to use some public event-based algorithms to identify different people and reconstruct images.

Concretely, E2VID [177] is utilized to carry out the visualization attack, while EV-Gait and 3DGraph-Gait are used to perform the recognition attack. E2VID provides a well-trained end-to-end neural network to reconstruct images from a stream of events. It firstly packs the events in a short period to a 3D tensor and then cooperates with several previously reconstructed images to generate a new image following the recurrent UNet architecture. This neural network is trained on synthetic data from an event simulator and performs well on real data. Its well-trained model weights have been released, and are utilized for attacks and evaluation in this work. EV-Gait and 3DGraph-Gait can identify different people using event streams without reconstructing grey-scale images. EV-Gait packs events to an event frame, and a ResNet with a fully connected neural network is employed to extract features and identify different people. 3DGraph-Gait models a stream of events as a 3D graph and applies OctreeGrid sampling strategy for downsampling. A 3D-graph is constructed according to the distance of x, y and t, and the GCN-based approach can work on the constructed graph for identification.

5.3 The Proposed Efficient Encryption Framework

Inspired by the encryption scheme for traditional images [132], we extend the approach of scrambling pixels' positions for shuffling both events' positions and their polarities. The encryption and decryption framework is illustrated as Fig. 5.4. The main idea of the proposed framework is that several pseudo-random sequences, generated using a chaotic map and updated over a period of time, can securely map an event's position and polarity to another position and polarity. Concretely, a two-dimensional chaotic map [132] firstly generates six pseudo-random sequences, which are employed to

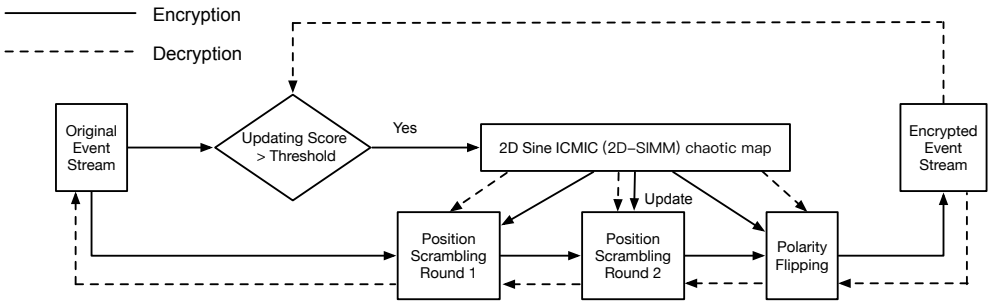


Figure 5.4: The flowchart of the proposed encryption and decryption framework. The updating score controls the updating speed of pseudo-random sequences, which are generated using chaotic mapping. The encryption (decryption) process consists of scrambling (restoring) positions and flipping (restoring) polarities.

randomly shuffle original events (scrambling position and flipping polarity), and an indicator, *updating score*, is designed to control the updating speed of the sequences according to the type of event cameras and the hardware configuration.

5.3.1 Pseudo-Random Sequence Generation

Chaotic systems are widely used techniques to generate pseudo-random sequences, which are applicable in image encryption. Due to its high sensitivity to initial conditions and parameters, chaotic systems can be employed to secure the generated pseudo-random sequence. For example, the sine map is a typical chaotic system, which is defined as:

$$x_{i+1} = \alpha \sin(x_i) \tag{5.1}$$

where α is a control parameter, and the x_{i+1} and x_i should be ranged in $[0, 1]$. Here, x_0, α and the iterations i jointly decide the current status of the chaotic system. However, a 1D chaotic system, such as a sine map, has few parameters and initial status, and the system status is relatively simple. To overcome such a disadvantage, high-dimensional chaotic maps have been proposed, but their computational complexity increases heavily compared with that of 1D chaotic systems. As a trade-off, two-dimensional (2D) chaotic maps can achieve better chaotic performance and introduce acceptable overhead.

A 2D chaotic map combines two different types of 1D chaotic maps. Besides the sine map, the iterative chaotic map with infinite collapse (ICMIC) [80] also shows

robust chaotic characteristics, expressed as:

$$x_{i+1} = \sin(\frac{\beta}{x_i}) \tag{5.2}$$

where $x_i \in (-1, 1)$, $x_0 \neq 0$ and $\beta \in (0, +\infty)$. β is also a control parameter. Based on these 1D chaotic maps, a two-dimensional Sine ICMIC modulation map (2D-SIMM) [132] is defined as:

$$\begin{cases} x_{i+1} = a\sin(\pi y_i)\sin(\frac{b}{x_i}) \\ y_{i+1} = a\sin(\pi x_{i+1})\sin(\frac{b}{y_i}) \end{cases} \tag{5.3}$$

where system parameters $a, b \in (0, +\infty)$. Here, a, b, x_0 and y_0 collectively describe the initial status of the chaotic system, and the iterations i decides the current status. a and b have been set as 1 and 5 respectively to generate pseudo-random sequences, because the system is a hyperchaotic map in this setting.

Since more than one pseudo-random sequences are required, to reduce the length of keys, the initial status of the system is decomposed to x_0(52 bits), y_0(52 bits), H(52 bits) and G(25 bits). The encryption key is a 306-bit string, including x_0, y_0 H and G_0-G_5, where x_0, y_0 and H are 52 bits each, and G_0-G_5 is 25 bits each. The sequences share the same x_0, y_0 and H, but have different G values. According to the designed encryption and decryption frame, six pseudo-random sequences are required. If different X, Y, H and G are set for each sequence, the length of the encryption key is more than 1000 bits. By balancing the length of the key and the security strength, this shared key approach has been designed. The initial status under condition G, x_0^G and y_0^G is donated as:

$$\begin{cases} x_0^G = (x_0 + GH) mod 1 \\ y_0^G = (y_0 + GH) mod 1 \end{cases} \tag{5.4}$$

Given that the width and height of the event cameras' resolution are M and N, we generate six pseudo-random sequences, whose lengths are N, M, N, M, $2N$ and $2M$, respectively, and number these sequences from r_1 to r_6.

5.3.2 Encryption and Decryption Algorithms

The generated pseudo-random sequences are utilized to scramble the positions of events and flip the corresponding polarities. During the encryption process, there are two rounds for scrambling and one round for flipping. The former focuses on changing the positions of events, and the latter modifies the polarities with the position shuffle.

Position Scrambling Round. The first round of scrambling utilizes the sequence r_1 and r_2, while r_3 and r_4 are used for the second round. Here, r_1^i denotes the ith element of the sequence r_1. The scrambling on y under r_1 is formulated as:

$$f_{r_1}(x, y) = (x, (y + r_1^x) \mod M)) \tag{5.5}$$

which means that an event at the position (x, y) moves r_1^x steps right-forward to $(x, y + r_1^x)$. If the changed y exceeds the boundary M, the movement will start from the $(x, 0)$ and end with $(x, (y + r_1^x) \mod M))$. Similarly, the scrambling on x under r_2 is formulated as:

$$f_{r_2}(x, y) = ((x + r_2^y) \mod N, y) \tag{5.6}$$

The scrambling in the x direction is conducted after the y direction scrambling. After completing the first x and y direction scrambling, spatially adjacent events will be distributed to different positions. However, one round is not enough to scramble all events thoroughly [132]. Thus, the second round scrambling is executed again using the same methods but based on r_3 and r_4.

Polarity Flipping Round. After scrambling the positions of events, polarity flipping is associated with the scrambled position and the original polarity. There are two steps to flip the polarity, which shuttle the position as well. The first step is to flip the polarity based on the scrambled y and the original polarity, while the second step flips it according to the scrambled y and the first step's result. Specifically, the r_5 is utilized for the first step, whose length is $2N$. An event, which is processed after position scrambling round and is located at (x, y, p), is associated with the $(p * N + y)$th

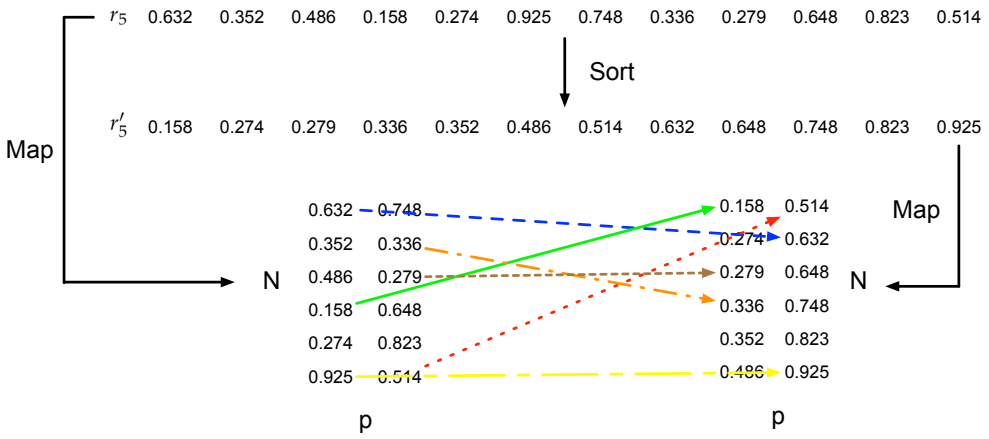


Figure 5.5: A polarity flipping example using r_5 to shuttle y and p

element of the r_5, where $x \in [0, M-1]$, $y \in [0, N-1]$ and $p \in 0, 1$. Then, a sort operation $\mathcal{S}$ is conducted on the r_5, denoted as $\mathcal{S}_{r_5}$. A new sequence r'_5 is acquired, and the original element, located at $p*N+y$, will be relocated at $l' = \mathcal{S}_{r_5}(p*N+y)$. l' is mapped to $(l' \mod N, \lfloor \frac{l'}{N} \rfloor)$. This transformation can be expressed as:

$$f_{r_5}(x, y, p) = (x, \mathcal{S}_{r_5}(p*N+y) \mod N, \lfloor \frac{\mathcal{S}_{r_5}(p*N+y)}{N} \rfloor)) \tag{5.7}$$

Similarly, the transformation of second step is conducted as:

$$f_{r_6}(x, y, p) = (\mathcal{S}_{r_6}(p*M+x) \mod M, y, \lfloor \frac{\mathcal{S}_{r_6}(p*M+x)}{M} \rfloor)) \tag{5.8}$$

An example of polarity flipping using r_5 is illustrated in Fig. 5.5. In this example, $(x, 0, 0)$ is mapped to $(x, 1, 1)$, which is marked as the blue line, while the orange line shows that $(x, 1, 1)$ is mapped to $(x, 3, 0)$.

The encryption algorithm can be summarized in Algorithm 1. The position scrambling is from line 1 to line 4, while the polarity flipping begins from line 5.

The processes of encryption and decryption are symmetrical because the operations in encryption are reversible. The decryption begins with recovering the polarity, and then each pixel is restored to its corresponding location. Algorithm 2 presents the decryption process. Similar to encryption, the sorted sequences r'_6 and r'_5 are generated from the pseudo-random sequences r_6 and r_5. Given the indexes of the sorted sequences, the inverse map $\mathcal{S}'$ returns the indexes of the pseudo-random sequences.

Algorithm 1: Encryption Algorithm

Input : $r_1, ..., r_6$: pseudo-random sequences
(t, x, y, p): an event
M, N: the width and height of the event camera

Output : (t, x', y', p'): an encrypted event

1 $y' = (y + r_1^x) \mod N$;
2 $x' = (x + r_2^{y'}) \mod M$;
3 $y' = (y + r_3^{x'}) \mod N$;
4 $x' = (x + r_4^{y'}) \mod M$;
5 $r'_5 = Sort(r_5)$;
6 Generate $\mathcal{S}_{r_5}$ which maps the index of each element in the original sequence of r_5 to the sorted r'_5;
7 $p_{temp} = \lfloor \frac{\mathcal{S}_{r_5}(p*N+y')}{N} \rfloor$;
8 $y' = \mathcal{S}_{r_5}(p * N + y') \mod N$;
9 $r'_6 = Sort(r_6)$;
10 Generate $\mathcal{S}_{r_6}$ which maps the index of each element in the original sequence of r_6 to the sorted r'_6;
11 $p' = \lfloor \frac{\mathcal{S}_{r_6}(p_{temp}*M+x')}{M} \rfloor$;
12 $x' = \mathcal{S}_{r_6}(p_{temp} * M + x') \mod M$;

From line 1 to line 8, the randomly flipping polarity is recovered back to its original polarity. Line 12 to line 15 shows the inverse transformation of position scrambling.

5.3.3 Pseudo-Random Sequence Updating

Although the encryption on the event stream prevents reconstructing grayscale images, events with the same position and polarity will be mapped to another same position and polarity. If constant pseudo-random sequences are used from the beginning to the end, the adversary can attack by mapping the original event stream to the encrypted one before the transmission. It is therefore necessary to update the pseudo-random sequences frequently. However, high updating frequency will reduce the efficiency of encryption , and thus it is an important factor to decide when to update the sequences. Here, we define an *updating score* to decide whether to update the pseudo-random sequences when a new event occurs, according to the type of a event camera and hardware configuration. Three parameters, the sensor's resolution ($N \times M$), the platform's processing speed (K), and the number of processed events since the last update (L), are considered affecting the updating score. The relationship between

Algorithm 2: Decryption Algorithm

Input : $r_1, ..., r_6$: pseudo-random sequences
(t, x', y', p'): an encrypted event
M, N: the width and height of the event camera

Output : (t, x, y, p): an event

$r'_6 = Sort(r_6)$;
Generate $\mathcal{S}'_{r_6}$ which maps the index of each element in the sorted sequence r'_6 to the original sequence r_6;
$p_{temp} = \lfloor \frac{\mathcal{S}'_{r_6}(p' * M + x')}{M} \rfloor$;
$x = \mathcal{S}_{r_6}(p' * M + x') \mod M$;
$r'_5 = Sort(r_5)$;
Generate $\mathcal{S}'_{r_5}$ which maps the index of each element in the sorted sequence of r'_5 to the original sequence r_5;
$p = \lfloor \frac{\mathcal{S}'_{r_5}(p_{temp} * N + y')}{N} \rfloor$;
$y = \mathcal{S}_{r_5}(p_{temp} * N + y') \mod N$;
$x = (x - r_4^y) \mod M$;
$y = (y - r_3^x) \mod N$;
$x = (x - r_2^y) \mod M$;
$y = (y - r_1^x) \mod N$;

these three values and the score is expressed as:

$$\text{Updating Score} = \log_{10}(\frac{L}{N \times M} \times K) \tag{5.9}$$

The large resolution leads to more relational mappings between unencrypted events and encrypted events. When the number of events processed is fixed, higher resolution implies that more relational mappings have not been involved, and thus the level of security is relatively higher. Moreover, the higher processing speed of the platform enables more frequent updating. For example, compared with Raspberry Pi, the cloud server can perform updates more frequently to achieve higher security. Finally, the effect of L is intuitive: with the more events processed since the last update, it is more desirable to perform the update.

5.4 Evaluation

In order to conduct evaluations in several perspectives, three public event-based datasets are utilised. The first one is the DAVIS event camera dataset [154], a real-world dataset captured in various scenarios such as labs, offices and campuses. This

dataset is published in 2017, which has been used for evaluations in a number of prior studies. DDD17 [29] is an end-to-end DAVIS driving dataset, which records events in driving scenarios on a highway and in a city under different weather conditions. In this study, qualitative and quantitative evaluations for visualization attacks are conducted using these datasets, as well as efficiency analysis and secret key analysis. DVS128-Gait is collected for human identification by their gaits. Because the number of events for each record is small, the visualization attack on this dataset did not succeed. Therefore, this dataset is employed to evaluate all tasks except for visualization attack prevention.

Due to the lack of previous encryption schemes for event cameras, we define two kinds of partial encryption schemes, inspired by the keyframe encryption method on video [6], to evaluate the proposed encryption framework. The first baseline encryption schemes is to apply streaming data encryption, Advanced Encryption Standard (AES), directly to a part of events. Under this scheme, encrypted events and unencrypted events can be easily distinguished, and attackers can utilize unencrypted events to perform attacks. We call this encryption algorithm the partial discarding algorithm. The second baseline is to apply the scrambling method to a part of events, which means that the unencrypted events cannot be identified from all events. We name this algorithm as the partial scrambling algorithm.

5.4.1 Evaluation for Visualization Attacks

To perform the evaluation for visualization attacks, the grayscale images are firstly reconstructed from the original stream using E2VID [177]. After encrypting the event stream, the same configuration of E2VID is utilized to generate encrypted grayscale images.

Qualitative Evaluation. The reconstructed image produced from the original event stream is shown as Fig. 5.6(a), while the corresponding image produced from the encrypted event stream is illustrated as Fig. 5.6(b). Since the architecture of E2VID is recurrent, an error in the previous reconstruction will be propagated to the next, and thus the generated image after encryption is nearly all dark. The reconstructed images corresponding to the partial discarding algorithm with different encryption

percentage values are illustrated as Fig 5.6(c)-(e). It can be observed that even when discarding 75% events, the outline of a person and a screen can still be recognized. The partial scrambling algorithm cannot effectively prevent the visualization attacks, shown in Fig 5.6(f)-(h), although it outperforms the partial discarding algorithm. Some shadows of the person and the screen can still be figured out. Compared with these two baselines, our proposed encryption algorithm achieves the best effectiveness.

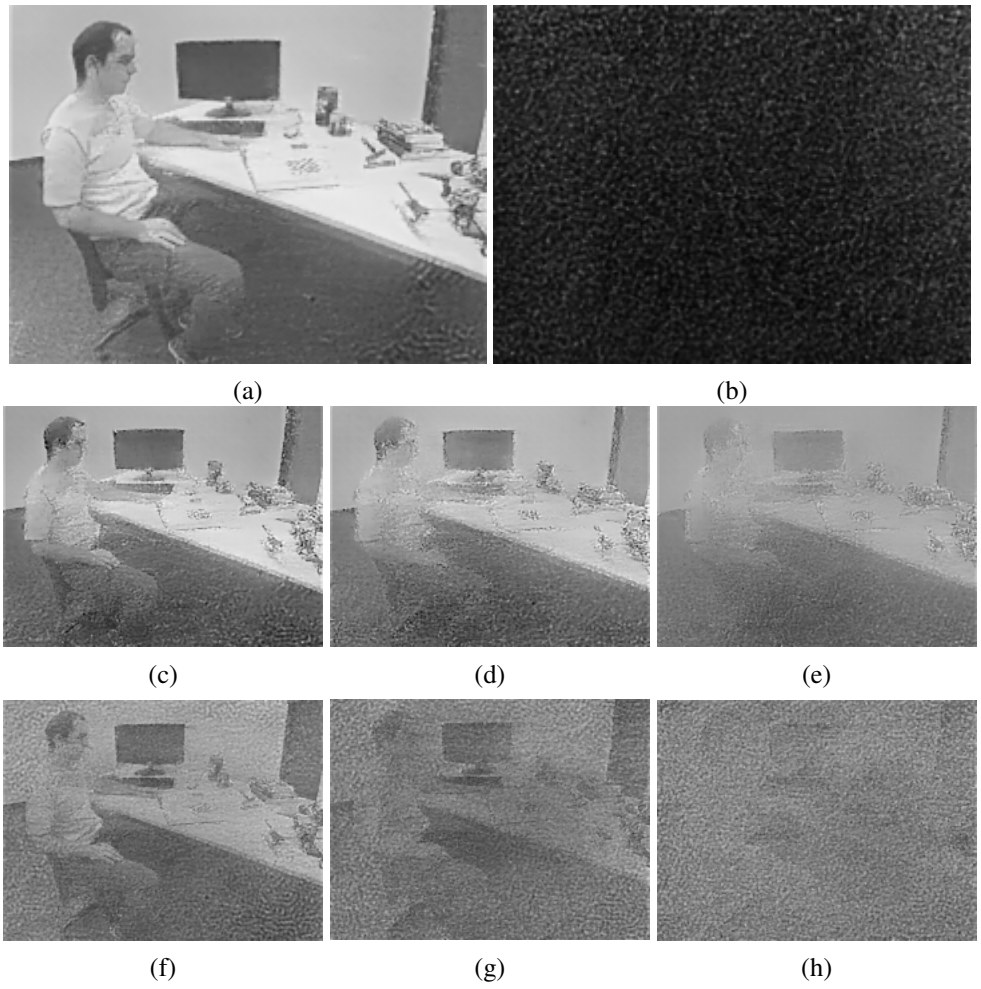
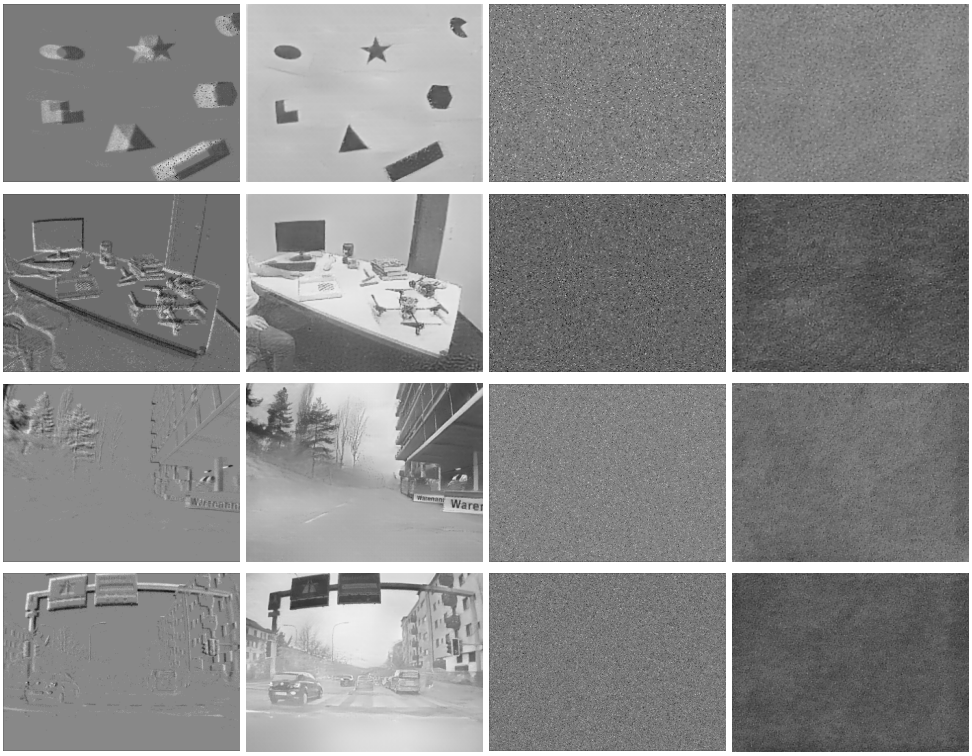


Figure 5.6: The reconstructed grayscale images of the same event stream under different conditions. (a) Image reconstructed from the original event stream. (b) Image reconstructed from encrypted events using our proposed framework. (c-e) Images reconstructed from 50%, 67%, and 75% encrypted events using the partial discarding algorithm, respectively. (f-h) Images reconstructed from 50%, 67%, and 75% encrypted events using the partial scrambling algorithm, respectively.

Here, more event frames and reconstructed images before and after using the proposed encryption algorithm are shown in Fig. 5.7. The first column presents the raw event frames, and the second column presents the reconstructed grayscale images

produced by the unencrypted data. The third column shows the event frames after encryption, while the last column displays the reconstructed images after encryption. The first two rows present the results using the DAVIS dataset, while the last two rows demonstrate the results using DDD17. As shown in Fig. 5.7, no outline or detail can be distinguished in the encrypted images (both event images and grayscale images), and the two images, before and after encryption, are totally different. It can therefore be concluded that the proposed encryption algorithm could be applied for diverse event cameras' hardware and scenarios.



(a) (b) (c) (d)

Figure 5.7: The qualitative evaluation of encryption. (a) The event images based on unencrypted events. (b) The reconstructed images based on unencrypted events. (c) The event images based on encrypted events using the proposed algorithm. (d) The reconstructed images based on encrypted events using the proposed algorithm.

Quantitative Evaluation. In order to quantitatively evaluate the proposed encryption algorithm's performance, we adopt Peak Signal to Noise Ratio (PSNR), Structural Similarity (SSIM), Unified Averaged Changed Intensity (UACI) and the Number of Pixel Changing Rate (NPCR) as metrics for comparing the images reconstructed from the original event stream and the encrypted one. Given the original reconstructed

image I and the reconstructed image I' based on the encrypted event stream, the definitions of these metrics are as follows:

$$PSNR = 10\log_{10}(\frac{M \times N}{\sum_{n,m}^{n \leq N, m \leq M}(I(n,m) - I'(m,n))^2}) \tag{5.10}$$

$$SSIM = \frac{(2\mu_I\mu_{I'} + C_1)(2\sigma_{II'} + C_2)}{(\mu_I^2 + \mu_{I'}^2 + C_1)(\sigma_I^2 + \sigma_{I'}^2 + C_2)} \tag{5.11}$$

$$UACI = \frac{1}{M \times N} \sum_{n,m}^{n \leq N, m \leq M} (I(n,m) - I'(m,n)) \tag{5.12}$$

$$PNCR = \frac{1}{M \times N} \sum_{n,m}^{n \leq N, m \leq M} \delta(I(n,m) - I'(m,n)) \tag{5.13}$$

The comparison is performed between the original reconstructed images and the processed images after the proposed encryption and the other two baselines, and the results are presented in Table 5.1 and Table 5.2. According to the results, the proposed encryption algorithm achieves the lowest values for PSNR and SSIM, and the highest for UACI and NPCR, compared with the baselines. The average PSNR value based on our algorithm is lower than 6.5, while that of other baselines exceed 10.5. Our algorithm also achieves the lowest SSIM values in all sequences. The UACI and NPCR of the proposed algorithm are the best for most of the sequences, and slightly lower than that of the partial scrambling encryption algorithm. The PSNR between the original reconstructed image and the reconstructed image after encryption with the outdoor running sequence is higher than that with other sequences, because most of the pixels in the reconstructed images with the outdoor running sequence are gray, which implies that their intensity value is close to 0.5. Compared with scrambling 95% of events, the proposed encryption algorithm additionally flips the polarity of events, making the encrypted events' polarities distributed uniformly in $\{+1, -1\}$. Under this mechanism, the pixel values of the reconstructed image after encryption are also close to 0.5. Although our algorithm does not achieve the best performance on PSNR, UACI and NPCR using the outdoor running sequence, the uniformly distribution characteristic can effectively destroy the original pixel value distribution, leading to

the best SSIM performance. Overall, our algorithm also outperforms others in the quantitative evaluation.

Dataset	Sequence	Ours		95% discarding		95% scrambling	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DAVIS	dynamic_6dof	**5.43**	**0.07**	14.60	0.53	13.47	0.13
	poster_6dof	**7.27**	**0.06**	15.26	0.21	14.19	0.10
	shapes_rotation	**5.98**	**0.11**	14.86	0.79	8.16	0.15
	outdoors_running	11.77	**0.09**	12.06	0.40	**10.82**	0.16
DDD17	rec1487779465	**5.17**	**0.07**	13.32	0.61	9.19	0.13
	rec1487839456	**4.02**	**0.03**	14.84	0.53	4.47	0.06
	rec1487609463	**3.97**	**0.07**	17.49	0.83	14.04	0.15

Table 5.1: The PSNR and SSIM results of the comparison between the original reconstructed images and the reconstructed images after using different encryption algorithms

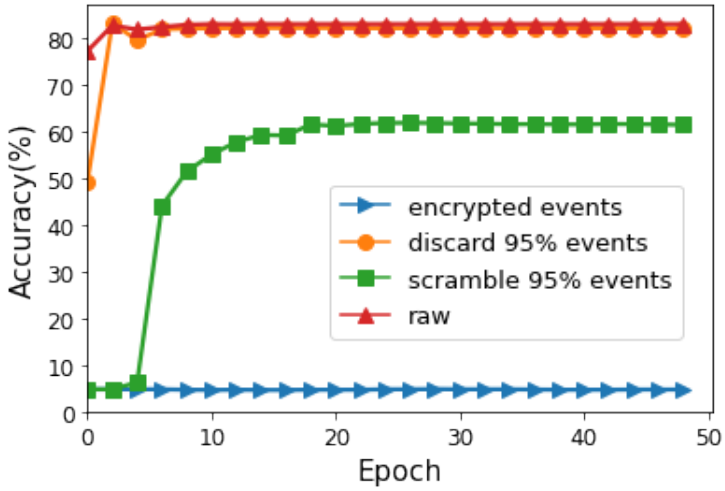
Dataset	Sequence	Ours		95% discarding		95% scrambling	
		UACI	NPCR	UACI	NPCR	UACI	NPCR
DAVIS	dynamic_6dof	**52.00**	**99.96**	15.10	98.94	18.78	99.55
	poster_6dof	**41.43**	**99.89**	14.19	99.13	16.07	99.25
	shapes_rotation	**50.03**	**99.99**	16.34	99.49	39.30	99.97
	outdoors_running	22.54	99.34	23.15	99.35	**25.22**	**99.56**
DDD17	rec1487779465	**54.35**	**99.95**	18.91	98.99	31.76	99.74
	rec1487839456	**61.35**	**99.99**	14.81	99.07	58.32	**99.99**
	rec1487609463	**64.30**	**99.99**	10.55	97.82	16.90	99.23

Table 5.2: The UACI and NPCR results of the comparison between the original reconstructed images and the reconstructed images after using different encryption algorithms

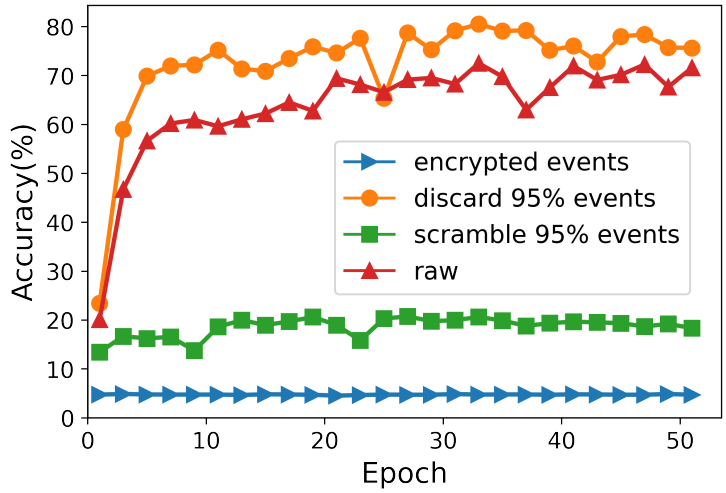
5.4.2 Evaluation for Recognition Attacks

For the recognition attacks, the accuracy is employed as the metric to quantitatively validate how the encryption algorithm prevents the deep learning-based identification attacks, including an event frame-based CNN and a sparse event-based GCN. The accuracy values with different encryption algorithms at different epochs are presented in Fig. 5.8.

Using the CNN approach, recognition attack can achieve 86% accuracy on original events and 81% accuracy on 95%-encrypted events using the partial discarding algorithm. Moreover, For 95%-encrypted events using the partial scrambling algorithm, the attack reaches approximately 60% accuracy. In contrast, based on the our proposed



(a) The recognition accuracy using CNN



(b) The recognition accuracy using GCN

Figure 5.8: The accuracy of recognition attacks using event frame-based CNN and sparse event-based GCN under different conditions

algorithm, only 5% identification is correct, equal to the random guess. It is clear that the performance of our algorithm in recognition attack is as good as its performance in visualization attack, because the CNN-based attack extracts features from the packed frames, which are similar to the reconstructed images. In conclusion, the proposed algorithm effectively prevent the CNN-based attack approach and outperforms other encryption algorithms.

The performance under the GCN based-attack depends on the events that are used to constructed the 3D graph. The attack on the events using the partial discarding algorithm can achieve about 73% accuracy before the 50th epochs, while that using

the partial scrambling algorithm can achieve about 20% accuracy. In contrast, with our proposed encryption algorithm, the attack achieves only 5% accuracy. In conclusion, the proposed algorithm successfully destroys the relationships among events in 3D spaces.

5.4.3 Secret Key Analysis

The space of the secret key is an important factor affecting the application security in real scenarios. A large keyspace can make it difficult for attackers to enumerate the possible keys. The secrete key, including x_0, y_0, H and Gs, is 306 bits in length, and thus, the keyspace is about 2^{306}. It is impossible to crack the key by the brute-force attacks in a reasonable duration. Apart from the keyspace, the sensitivity of the encryption algorithm to the secrete key also affects the usage. The original event frame and the decrypted event frames are illustrated in Fig. 5.9. The correct key, K_0, can be used to decrypt the encrypted event frame. $K_1 - K_6$ are the incorrect keys, which are produced by modifying the last bits of Gs. The images produced by the decrypted events with slightly different keys, K_1 to K_6, are totally different from the event image using the right key. Therefore, the proposed encryption algorithm is sensitive enough to the secrete key.

Dataset	Sequence	Events/Sec	Updating Score				
			7.2	7.6	8.0	8.4	8.8
Gait 128*128	2-1	17034	0.549	0.261	0.114	0.179	0.115
	2-2	14036	0.513	0.242	0.177	0.115	0.104
	2-3	11545	0.535	0.252	0.183	0.110	0.111
DAVIS 240*180	shape_6dof	299375	0.463	0.200	0.113	0.055	0.039
	shapes_rotation	385438	0.511	0.222	0.097	0.056	0.039
	shapes_translation	289400	0.461	0.200	0.099	0.055	0.039
DDD17 346*260	rec1487609463	153625	0.563	0.243	0.127	0.069	0.050
	rec1487779465	474789	0.559	0.245	0.124	0.075	0.051
	rec1487839456	680153	0.564	0.246	0.120	0.072	0.051

Table 5.3: Time (μs) spent on encrypting one event on Raspberry Pi (K=36,694,061) using different updating scores and event cameras with different resolutions

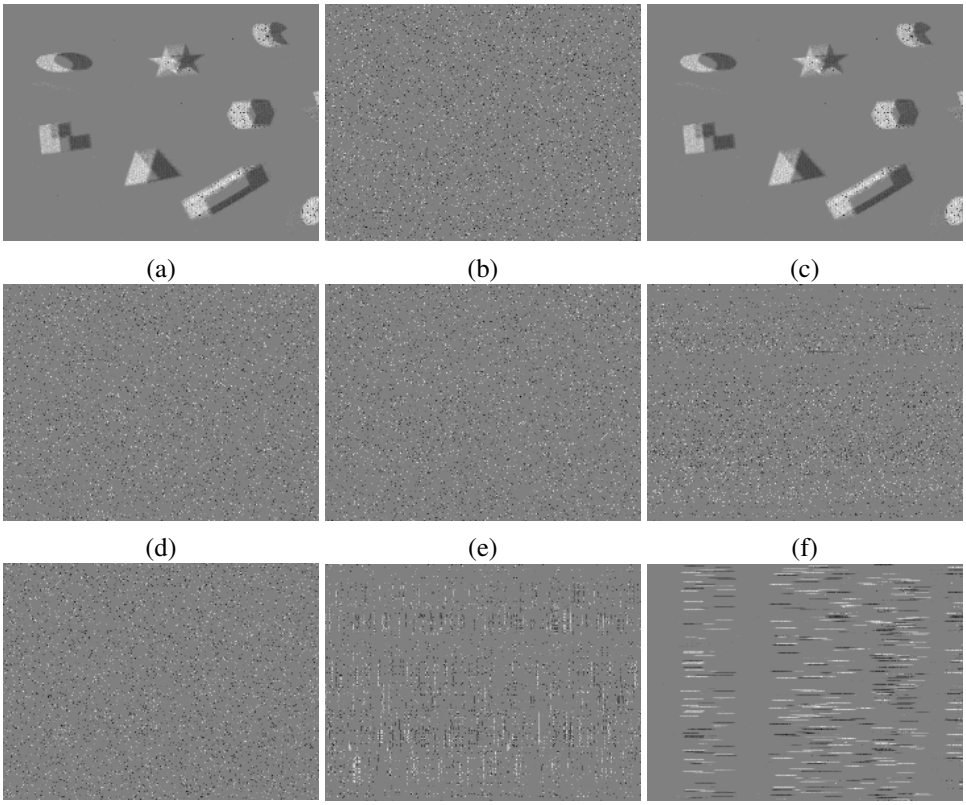


Figure 5.9: The sensitivity study on the secret keys. (a) The original event image. (b) The event image after encryption. (c) The event image after decryption with the correct secrete key K_0. (d-i) The event images after decryption with incorrect secrete keys K_1, K_2, K_3, K_4, K_5 and K_6, respectively.

5.4.4 Efficiency Analysis

The proposed encryption has been performed on different platforms for evaluation. For outdoor surveillance purpose, the real-time requirement on a resource-constrained platform is compulsory. The communication between the platforms of event cameras and the servers requires both encryption and decryption. These experiments are conducted on a Raspberry Pi, a desktop server and a cloud server. The Raspberry Pi (4b edition) is equipped with 8GB RAM. The desktop server is equipped with an Intel Core i9-9900K@3.6GHz processor with 32GB RAM, and the cloud server is equipped with an Intel Xeon Platinum 8269CY@2.5GHz and 64GB RAM.

The time (μs) spent on encrypting each event is measured and listed in Table 5.3, Table 5.4 and Table 5.5. When the updating score is 7.2, the average time to encrypt one event is less than 0.60 μs on Raspberry Pi. When the updating score increases to

Dataset	Sequence	Events/Sec	Updating Score				
			7.2	7.6	8.0	8.4	8.8
Gait 128*128	2-1	17034	0.619	0.180	0.086	0.039	0.024
	2-2	14036	0.415	0.167	0.080	0.038	0.023
	2-3	11545	0.418	0.175	0.084	0.041	0.023
DAVIS 240*180	shape_6dof	299375	0.398	0.163	0.068	0.031	0.016
	shapes_rotation	385438	0.398	0.163	0.069	0.032	0.016
	shapes_translation	289400	0.398	0.163	0.068	0.031	0.016
DDD17 346*260	rec1487609463	153625	0.419	0.171	0.072	0.033	0.017
	rec1487779465	474789	0.418	0.171	0.072	0.033	0.018
	rec1487839456	680153	0.420	0.171	0.073	0.035	0.018

Table 5.4: Time (μs) spent on encrypting one event on desktop server (K=253,725,220) using different updating scores and event cameras with different resolutions

Dataset	Sequence	Events/Sec	Updating Score				
			7.2	7.6	8.0	8.4	8.8
Gait 128*128	2-1	17034	0.406	0.173	0.069	0.034	0.023
	2-2	14036	0.413	0.170	0.075	0.032	0.023
	2-3	11545	0.416	0.169	0.068	0.032	0.024
DAVIS 240*180	shape_6dof	299375	0.400	0.161	0.067	0.029	0.014
	shapes_rotation	385438	0.399	0.161	0.067	0.029	0.014
	shapes_translation	289400	0.400	0.162	0.067	0.029	0.014
DDD17 346*260	rec1487609463	153625	0.406	0.165	0.069	0.030	0.015
	rec1487779465	474789	0.406	0.166	0.069	0.030	0.015
	rec1487839456	680153	0.411	0.166	0.069	0.031	0.015

Table 5.5: Time (μs) spent on encrypting one event on cloud server (K=152,031,121) using different updating scores and event cameras with different resolutions

8.4, encrypting one event only consumes about 0.20 μs on Raspberry Pi, and only 0.04 μs on both desktop and cloud servers. The numbers of processed events per second with different updating scores are demonstrated in Fig. 5.10. On the Raspberry Pi, more than tens of millions of events can be encrypted per second. These experimental results indicate that the proposed encryption framework works efficiently on various platforms, including resource-constrained devices.

5.5 Summary

Gait recognition with event cameras has demonstrated some advantages, such as energy consumption and efficiency, and competing accuracy with traditional RGB cameras. As event cameras generate a stream of impulse-like events (rather than

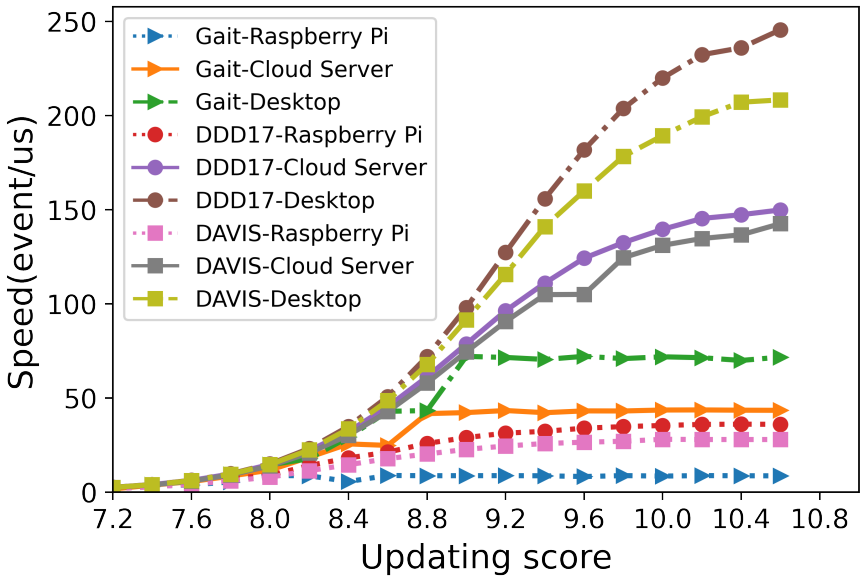


Figure 5.10: The relationship between the updating score and the number of events processed per second (in μs) using different event cameras on various platforms

synchronized image frames generated by traditional RGB-based vision sensors), prior work assumed they are privacy-preserving. However, applying event cameras for sensitive tasks or on sensitive scenarios should carefully consider their security and users' privacy. Major contributions in this chapter are summarized as follows:

- Firstly, we have identified and investigated major privacy issues related to event cameras, and proposed a threat model that reveals serious security risks of event camera-based applications. Given the capabilities and knowledge defined in the model, adversaries can perform visualization and recognition attacks to achieve their objectives.

- Secondly, we have proposed and designed an efficient encryption framework for protecting event cameras' data against visualization and recognition attacks. The framework incorporates a 2D chaotic mapping-based encryption algorithm and a secret key updating mechanism based on an updating score.

- Finally, extensive experiments have been conducted, which demonstrated that the proposed framework can effectively and efficiently protect events of event cameras against grayscale image reconstruction and human identification. Specifically, the proposed encryption framework can be efficiently executed on

resource-constrained devices.

We expect that privacy and security issues related to event cameras would attract increasing attentions and interests from both academia and industry in the upcoming years, given the promising advantages and applications of event cameras. We will be continuously working in this emerging field, exploring more potential threats to event cameras, and proposing novel approaches and solutions to meet the challenges. Furthermore, some dedicated privacy-preserving approaches can be designed for different tasks, such as gait recognition and object detection, protecting the privacy and keeping high accuracy for target tasks.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Nowadays, our daily lives have been benefiting from various computer vision technologies, from robotic vacuums to autonomous driving vehicles. Recognising objects or people is a fundamental task for more complex tasks, such as manipulating objects and tracking. Some biometric characteristics, such as face, iris and gait, have been employed for human identification. Compared with the widely used face recognition, gait-based approaches can work remotely and are unlikely to leak sensitive information that can be used for some high-level security systems, such as the biometric passport. Many kinds of sensors have been utilised for gait recognition, such as standard cameras, infrared cameras, floor sensors and inertial sensors. However, devices for gait recognition should be available any time and are able to respond quickly. The emerging of event cameras exactly satisfies such requirements. The event camera is a bio-inspired sensor with a high temporal resolution, wide dynamic range, and low energy consumption. Although such advantages make it suitable for surveillance, there is no dedicated solution that can process the outputs of this new camera, which makes it difficult to solve gait recognition tasks directly.

In this thesis, we have focused on gait recognition with event cameras. As the outputs of event cameras are not similar to that of standard cameras, two event-based gait datasets, namely DVS128-Gait and EV-CASIA-B, are firstly constructed for further learning and evaluation. EV-Gait, a CNN-based model, has been proposed for

human identification using the event stream, which packs the stream into an image-like frame and utilises a ResNet-like structure to extract features and recognise targets. Furthermore, 3DGraph-Gait, a GCN-based model, has been proposed, which can effectively capture the spatiotemporal features from events and perform well by either using the whole event stream or using only several hundred events produced in several milliseconds. In addition, a dedicated encryption framework for events, namely EV-Encryp, has been proposed, which can secure the applications of event cameras while introducing little overheads for data transmission and storage on various platforms. These contributions have enabled event cameras to perform gait recognition tasks in real-world scenarios.

6.1.1 Event-Based Datasets for Gait Recognition

In order to provide training samples and to perform quantitative evaluation, event-based gait dataset is required. We have produced two gait recognition datasets, namely DVS128-Gait and EV-CASIA-B. DVS128-Gait is captured in real-world settings, where 21 volunteers were involved for more than three weeks. This dataset can be used to train a gait recognition neural network, which can serve as the benchmark to evaluate the performance of gait recognition approaches based on event cameras. On the other hand, CASIA-B is the most widely used traditional-camera-based dataset for gait recognition, including data records generated from various angles of view. Based on this dataset, we have employed a DVS128 to record the playbacks of the video on a screen and generated the EV-CASIA-B dataset. Although EV-CASIA-B is synthesized, it provides a benchmark for comparison with traditional cameras. With the help of DVS128-Gait and EV-CASIA-B, some models for gait recognition are trained and evaluated.

6.1.2 Image-Based Convolution Enabling Gait Recognition with Event Cameras

Because the generation mechanism of events is related to intensity changes, an image-like representation method is utilised. Given CNN's capability in extracting features from RGB images, the proposed EV-Gait utilises a CNN structure to extract features

from the packed frames. Prior to feature extraction, a noise cancellation method with gait recognition is applied, which removes noise by enforcing motion consistency. The accuracy of EV-Gait enables the possibility of using event cameras to solve gait recognition tasks, which reaches more than 80% accuracy on DVS128-Gait. According to the comparison before and after noise cancellation, noise in event streams has negative effects on gait recognition, and the proposed cancellation can effectively filter the noise. EV-Gait brings traditional feature extraction methods directly to event streams and completes the gait recognition task.

6.1.3 Graph-Based Convolution Capturing Spatiotemporal Features for Event Stream

Events sparsely distribute in the spatiotemporal domain, which is similar to the point cloud in a 3D space. Although CNN achieves good performance for feature extraction on packed frames, some spatiotemporal features have been lost during the packing process. A GCN combined with 3D coordinate information can be utilised to extract 3D features. The proposed 3DGraph-Gait utilises such GCN to generate spatiotemporal features for gait recognition. Because the number of events generated in one second is large, some sampling approaches have been designed to enable the neural network to deal with the processed events. In addition, this network also shows a powerful capability to extract features using a limited number of events. By using only several hundred events, the ensemble model can achieve more than 90% accuracy. Compared with features captured by CNN, the extracted spatiotemporal features by GCN are more valuable for gait recognition.

6.1.4 Event-Oriented Encryption Framework

As the event camera is an emerging vision sensor, no prior work had investigated its security issues. However, security and privacy may seriously affect the application of event cameras for gait recognition. Firstly, gait recognition is a sensitive task, and the lack of protection on event streams limits the application in some security-critical scenarios. Besides gait information, other information that is irrelevant to the gait, such as environments and surroundings, may cause potential privacy leakage. EV-Encryp,

an efficient encryption framework for event cameras, has been proposed to cope with such challenges. The number of events generated in a short period is large, and thus the two-dimensional chaotic map approach is utilised to generate a pseudo-random sequence efficiently. Based on the sequence, the position and polarity of events are scrambled and shuffled. Furthermore, this encryption framework can work efficiently on different platforms, based on a specific mechanism with an updating score. Under the protection by this encryption framework, event cameras can securely work for gait recognition.

6.2 Future Work

Following our existing achievements on gait recognition with event cameras, several research issues have been identified for further exploration. Different kinds of representation methods and neural networks have different capabilities to present and extract features from the event stream. If different features, such as static features and dynamic features, are considered to be used jointly, how to choose representations and networks and how to fuse the extracted features are valuable issues, which may boost the accuracy of gait recognition for event cameras. Furthermore, a universal gait recognition approach can be explored for both standard cameras and event cameras. There are many existing approaches for standard camera-based gait recognition, and our designed gait recognition solutions are dedicated for event cameras. These two modalities may share similar gait recognition features, which can be used as a universal describer to link the traditional gait recognition and event-based gait recognition. In addition, the privacy issues of event cameras deserve further exploration. The encryption is generally for all scenarios, and other privacy-preserving approaches that keep gait-related features and other hidden visual features can be further devised for event cameras.

6.2.1 Static Features and Dynamic Features Fusion for Gait Recognition

The features extracted from both packed frames and constructed 3D graphs describe dynamic patterns of gait. Some static features, such as the distance between a person's head and leg, also can be used to express the special characteristics of a particular person. Combing dynamic features with static features may improve the accuracy of gait recognition, but there are several critical problems associated with static feature extraction and fusion. The first problem is what kind of representation can most effectively express the gait-related static features of event streams. Both packed frames and 3D graphs are considered to present the dynamic information, but whether these two representation approaches can be further used to demonstrate static information could be explored. How many events most clearly show the static information is the second problem. There is no fixed exposure time for event cameras and no full image with all information. A single event lacks enough information, and events in a long period will cause unclear packed frames and redundant spatiotemporal features for graphs. An event stream with a proper length may benefit the static feature extraction. After extracting static features, the last problem is what kind of neural network can be utilised to fuse the static and dynamic features. Static features can help recognise people quickly, and combining these two features may improve the accuracy when using events in a short period.

6.2.2 Universal Gait Recognition for Standard Cameras and Event Cameras

Event cameras are a new kind of vision sensors and cannot be deployed widely quickly. In most scenarios, they act as an important supplement to traditional cameras. How to make event cameras cooperatively working with standard cameras will be an essential research problem in the upcoming years. For gait recognition with event cameras, can the standard camera-based algorithms and neural networks work for event cameras without being retrained or deep modification? If possible, a universal framework can work for both standard cameras and event cameras, which implies

that gait recognition can be conducted across different platforms. There are some image-oriented representation approaches for gait recognition, such as silhouette and GEI. That can be treated as the intermediate outcome of gait recognition. The feature extraction and classification components can be shared between standard and event cameras if they are possible to construct such intermediate results using event streams. Therefore, the key problem for building a universal gait recognition framework is proposing approaches to generating similar features between videos and event streams.

6.2.3 Event Camera Oriented Privacy-Preserving for Recognition

Encryption-based privacy protection approaches provide a general way to prevent illegal and unauthorised access. However, it introduces additional overhead for encryption and decryption, and cannot deal with privacy at different levels. An event-based privacy-preserving approach can be designed for recognition to overcome these limitations, which should satisfy the following requirements. Firstly, less overhead should be introduced at the same security level. Encryption-based approaches include two operations, encryption and decryption. If an irreversible operation that keeps relevant features and hidden privacy-related features can be designed, it will be more efficient than encryption. Secondly, only target-oriented features should be preserved and extracted. i.e., a proper privacy-preserving scheme can scramble the distribution of events in the spatiotemporal domain so that recognition-related features are preserved and other features are destroyed. Finally, the processed event streams cannot be restored. Because a single event holds limited information, the privacy-preserving approaches should focus on hiding the features of event streams rather than the information of a single event. As long as the main features of event streams cannot be restored, the privacy-preserving approaches for event cameras are effective.

Bibliography

[1] Pointmatcher library tutorial. https://libpointmatcher.readthedocs.io/en/latest/#tutorials.

[2] Pytorch geometric documentation. https://pytorch-geometric.readthedocs.io/en/latest/.

[3] Inivation dvs128. https://inivation.com/support/hardware/dvs128/.

[4] Intel up board. https://up-board.org/.

[5] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[6] Iskender Agi and Li Gong. An empirical study of secure mpeg video transmissions. In *Proceedings of internet society symposium on network and distributed systems security*, pages 137–144. IEEE, 1996.

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

[8] Munif Alotaibi and Ausif Mahmood. Improved gait recognition based on specialized deep convolutional neural network. *Computer Vision and Image Understanding*, 164:103–110, 2017.

[9] Ignacio Alzugaray and Margarita Chli. Asynchronous corner detection and tracking for event cameras in real time. *IEEE Robotics and Automation Letters*, 3(4):3177–3184, 2018.

[10] Weizhi An, Shiqi Yu, Yasushi Makihara, Xinhui Wu, Chi Xu, Yang Yu, Rijun Liao, and Yasushi Yagi. Performance evaluation of model-based gait on multi-view very large population database with pose sequences. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2(4):421–430, 2020.

[11] Saeed Anwar, Salman Khan, and Nick Barnes. A deep journey into super-resolution: A survey. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.

[12] Shai Avidan and Moshe Butman. Blind vision. In *European conference on computer vision*, pages 1–13. Springer, 2006.

[13] Shai Avidan, Ariel Elbaz, Tal Malkin, and Ryan Moriarty. Oblivious image matching. In *Protecting Privacy in Video Surveillance*, pages 49–64. Springer, 2009.

[14] Michal Balazia and Petr Sojka. You are how you walk: Uncooperative mocap gait identification for video surveillance with incomplete and noisy data. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 208–215. IEEE, 2017.

[15] R Baldwin, Mohammed Almatrafi, Vijayan Asari, and Keigo Hirakawa. Event probability mask (epm) and event denoising convolutional neural network (edncnn) for neuromorphic cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1701–1710, 2020.

[16] Patrick Bardow, Andrew J Davison, and Stefan Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 884–892, 2016.

[17] Souptik Barua, Yoshitaka Miyatani, and Ashok Veeraraghavan. Direct face

detection and video reconstruction from event cameras. In *2016 IEEE winter conference on applications of computer vision (WACV)*, pages 1–9. IEEE, 2016.

[18] Khalid Bashir, Tao Xiang, and Shaogang Gong. Gait recognition using gait entropy image. 2009.

[19] Ganbayar Batchuluun, Hyo Sik Yoon, Jin Kyu Kang, and Kang Ryoung Park. Gait-based human identification by combining shallow convolutional neural network-stacked long short-term memory and deep convolutional neural network. *IEEE Access*, 6:63164–63186, 2018.

[20] Francesco Battistone and Alfredo Petrosino. Tglstm: A time based graph deep learning approach to gait recognition. *Pattern Recognition Letters*, 126: 132–138, 2019.

[21] Alex I Bazin and Mark S Nixon. Probabilistic combination of static and dynamic gait features for verification. In *Biometric Technology for Human Identification II*, volume 5779, pages 23–30. International Society for Optics and Photonics, 2005.

[22] Chiraz BenAbdelkader, Ross Cutler, and Larry Davis. Stride and cadence as a biometric in automatic person identification and verification. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 372–377. IEEE, 2002.

[23] Ryad Benosman, Sio-Hoï Sio-Hoï Ieng, Paul Rogister, and Christoph Posch. Asynchronous event-based hebbian epipolar geometry. *IEEE Transactions on Neural Networks*, 22(11):1723–1734, 2011.

[24] Ryad Benosman, Sio-Hoi Ieng, Charles Clercq, Chiara Bartolozzi, and Mandyam Srinivasan. Asynchronous frameless event-based optical flow. *Neural Networks*, 27:32–37, 2012.

[25] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi. Event-based visual flow. *IEEE Trans. Neural Netw. Learning Syst.*, 25(2):407–417, 2014.

[26] Raphael Berner, Christian Brandli, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. A 240× 180 10mw 12us latency sparse-output vision sensor for mobile applications. In *VLSI Circuits (VLSIC), 2013 Symposium on*, pages C186–C187. IEEE, 2013.

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

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

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

[30] Dmitri Bitouk, Neeraj Kumar, Samreen Dhillon, Peter Belhumeur, and Shree K Nayar. Face swapping: automatically replacing faces in photographs. In *ACM SIGGRAPH 2008 papers*, pages 1–8. 2008.

[31] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on pattern analysis and machine intelligence*, 23(3):257–267, 2001.

[32] Aaron F Bobick and Amos Y Johnson. Gait recognition using static, activity-specific parameters. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[33] Nikolaos V Boulgouris, Konstantinos N Plataniotis, and Dimitrios Hatzinakos. Gait recognition using dynamic time warping. In *IEEE 6th Workshop on Multimedia Signal Processing, 2004.*, pages 263–266. IEEE, 2004.

[34] Enrico Calabrese, Gemma Taverni, Christopher Awai Easthope, Sophie Skriabine, Federico Corradi, Luca Longinotti, Kynan Eng, and Tobi Delbruck.

Dhp19: Dynamic vision sensor 3d human pose dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

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

[36] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. Attention mechanisms for object recognition with event-based cameras. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1127–1136. IEEE, 2019.

[37] Andrea Censi and Davide Scaramuzza. Low-latency event-based visual odometry. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 703–710. IEEE, 2014.

[38] Aaron Chadha, Yin Bi, Alhabib Abbas, and Yiannis Andreopoulos. Neuromorphic vision sensing for cnn-based action recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7968–7972. IEEE, 2019.

[39] Hanqing Chao, Yiwei He, Junping Zhang, and Jianfeng Feng. Gaitset: Regarding gait as a set for cross-view gait recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8126–8133, 2019.

[40] Changhong Chen, Jimin Liang, Heng Zhao, Haihong Hu, and Jie Tian. Frame difference energy image for gait recognition with incomplete silhouettes. *Pattern Recognition Letters*, 30(11):977–984, 2009.

[41] Guang Chen, Hu Cao, Canbo Ye, Zhenyan Zhang, Xingbo Liu, Xuhui Mo, Zhongnan Qu, Jörg Conradt, Florian Röhrbein, and Alois Knoll. Multi-cue event information fusion for pedestrian detection with neuromorphic vision sensors. *Frontiers in neurorobotics*, 13:10, 2019.

[42] Guang Chen, Peigen Liu, Zhengfa Liu, Huajin Tang, Lin Hong, Jinhu Dong, Jörg Conradt, and Alois Knoll. Neuroaed: Towards efficient abnormal event detection in visual surveillance with neuromorphic vision sensor. *IEEE Transactions on Information Forensics and Security*, 16:923–936, 2020.

[43] Junming Chen, Jingjing Meng, Xinchao Wang, and Junsong Yuan. Dynamic graph cnn for event-camera based gesture recognition. In *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2020.

[44] Shoushun Chen and Menghan Guo. Live demonstration: Celex-v: a 1m pixel multi-mode event-based sensor. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1682–1683. IEEE, 2019.

[45] Franco Chiaraluce, Lorenzo Ciccarelli, Ennio Gambi, Paola Pierleoni, and Maurizio Reginelli. A new chaotic algorithm for video encryption. *IEEE Transactions on Consumer Electronics*, 48(4):838–844, 2002.

[46] Jonghyun Choi, Kuk-Jin Yoon, et al. Learning to super resolve intensity images from events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2768–2776, 2020.

[47] Xavier Clady, Sio-Hoi Ieng, and Ryad Benosman. Asynchronous event-based corner detection and matching. *Neural Networks*, 66:91–106, 2015.

[48] Robert T Collins, Ralph Gross, and Jianbo Shi. Silhouette-based human identification from body shape and gait. In *Proceedings of fifth IEEE international conference on automatic face gesture recognition*, pages 366–371. IEEE, 2002.

[49] Jörg Conradt, Matthew Cook, Raphael Berner, Patrick Lichtsteiner, Rodney J Douglas, and T Delbruck. A pencil balancing robot using a pair of aer dynamic vision sensors. In *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pages 781–784. IEEE, 2009.

[50] Matthew Cook, Luca Gugelmann, Florian Jug, Christoph Krautz, and Angelika

Steger. Interacting maps for fast visual interpretation. In *The 2011 International Joint Conference on Neural Networks*, pages 770–776. IEEE, 2011.

[51] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004.

[52] David Cunado, Mark S Nixon, and John N Carter. Using gait as a biometric, via phase-weighted magnitude spectra. In *International conference on audio-and video-based biometric person authentication*, pages 93–102. Springer, 1997.

[53] David Cunado, Mark S Nixon, and John N Carter. Automatic extraction and description of human gait models for recognition purposes. *Computer vision and image understanding*, 90(1):1–41, 2003.

[54] Shaveta Dargan and Munish Kumar. A comprehensive survey on the biometric recognition systems based on physiological and behavioral modalities. *Expert Systems with Applications*, 143:113114, 2020.

[55] John Daugman. How iris recognition works. In *The essential guide to image processing*, pages 715–739. Elsevier, 2009.

[56] T Delbruck, Michael Pfeiffer, Raphaël Juston, Garrick Orchard, Elias Müggler, Alejandro Linares-Barranco, and MW Tilden. Human vs. computer slot car racing using an event and frame-based davis vision sensor. In *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, pages 2409–2412. IEEE, 2015.

[57] Tobi Delbruck and Manuel Lang. Robotic goalie with 3 ms reaction time at 4% cpu load using event-based dynamic vision sensor. *Frontiers in neuroscience*, 7:223, 2013.

[58] Jeffrey Delmerico, Titus Cieslewski, Henri Rebecq, Matthias Faessler, and Davide Scaramuzza. Are we ready for autonomous drone racing? the uzh-fpv drone racing dataset. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6713–6719. IEEE, 2019.

[59] Mohammad Derawi and Patrick Bours. Gait and activity recognition using commercial phones. *computers & security*, 39:137–144, 2013.

[60] Marcin Derlatka and Mikhail Ihnatouski. Decision tree approach to rules extraction for human gait analysis. In *International Conference on Artificial Intelligence and Soft Computing*, pages 597–604. Springer, 2010.

[61] Bowen Du, Chris Xiaoxuan Lu, Xuan Kan, Kai Wu, Man Luo, Jianfeng Hou, Kai Li, Salil Kanhere, Yiran Shen, and Hongkai Wen. Hydradoctor: real-time liquids intake monitoring by collaborative sensing. In *Proceedings of the 20th International Conference on Distributed Computing and Networking*, pages 213–217, 2019.

[62] Bowen Du, Weiqi Li, Zeju Wang, Manxin Xu, Tianchen Gao, Jiajie Li, and Hongkai Wen. Event encryption for neuromorphic vision sensors: Framework, algorithm, and evaluation. *Sensors*, 21(13):4320, 2021.

[63] Peiqi Duan, Zihao W Wang, Xinyu Zhou, Yi Ma, and Boxin Shi. Eventzoom: Learning to denoise and super resolve neuromorphic events. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12824–12833, 2021.

[64] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, 2001.

[65] Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 869–877, 2018.

[66] Thomas Finateu, Atsumi Niwa, Daniel Matolin, Koya Tsuchimoto, Andrea Mascheroni, Etienne Reynaud, Pooria Mostafalu, Frederick Brady, Ludovic Chotard, Florian LeGoff, et al. 5.10 a 1280$\times$ 720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86 μm pixels, 1.066 geps

readout, programmable event-rate controller and compressive data-formatting pipeline. In *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*, pages 112–114. IEEE, 2020.

[67] Yang Fu, Yunchao Wei, Yuqian Zhou, Honghui Shi, Gao Huang, Xinchao Wang, Zhiqiang Yao, and Thomas Huang. Horizontal pyramid matching for person re-identification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8295–8302, 2019.

[68] Davrondzhon Gafurov, Kirsi Helkala, and Torkjel Søndrol. Biometric gait authentication using accelerometer sensor. *J. comput.*, 1(7):51–59, 2006.

[69] Guillermo Gallego and Davide Scaramuzza. Accurate angular velocity estimation with an event camera. *IEEE Robotics and Automation Letters*, 2(2): 632–639, 2017.

[70] Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *IEEE Int. Conf. Comput. Vis. Pattern Recog.(CVPR)*, volume 1, 2018.

[71] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *arXiv preprint arXiv:1904.08405*, 2019.

[72] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[73] Arren Glover and Chiara Bartolozzi. Robust visual tracking with a freely-moving event camera. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3769–3776. IEEE, 2017.

[74] Michela Goffredo, Imed Bouchrika, John N Carter, and Mark S Nixon. Self-calibrating view-invariant gait biometrics. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(4):997–1008, 2010.

[75] Ralph Gross and Jianbo Shi. The cmu motion of body (mobo) database. 2001.

[76] Zhi-Hong Guan, Fangjun Huang, and Wenjie Guan. Chaos-based image encryption algorithm. *Physics Letters A*, 346(1-3):153–157, 2005.

[77] Qing Guo and Dan Jiang. Method for walking gait identification in a lower extremity exoskeleton based on c4. 5 decision tree algorithm. *International Journal of Advanced Robotic Systems*, 12(4):30, 2015.

[78] Jinguang Han and Bir Bhanu. Individual recognition using gait energy image. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (2):316–322, 2006.

[79] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[80] Di He, Chen He, Ling-Ge Jiang, Hong-wen Zhu, and Guang-rui Hu. Chaotic characteristics of a one-dimensional iterative map with infinite collapses. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 48(7):900–906, 2001.

[81] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[82] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang. Face recognition using laplacianfaces. *IEEE transactions on pattern analysis and machine intelligence*, 27(3):328–340, 2005.

[83] Yiwei He, Junping Zhang, Hongming Shan, and Liang Wang. Multi-task gans for view-specific feature learning in gait recognition. *IEEE Transactions on Information Forensics and Security*, 14(1):102–113, 2018.

[84] Martin Hofmann, Sebastian Bachmann, and Gerhard Rigoll. 2.5 d gait biometrics using the depth gradient histogram energy image. In *2012 IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 399–403. IEEE, 2012.

[85] Martin Hofmann, Jürgen Geiger, Sebastian Bachmann, Björn Schuller, and Gerhard Rigoll. The tum gait from audio, image and depth (gaid) database: Multimodal recognition of subjects and traits. *Journal of Visual Communication and Image Representation*, 25(1):195–206, 2014.

[86] Yuhuang Hu, Hongjie Liu, Michael Pfeiffer, and Tobi Delbruck. Dvs benchmark datasets for object tracking, action recognition, and object recognition. *Frontiers in neuroscience*, 10:405, 2016.

[87] Yuhuang Hu, Jonathan Binas, Daniel Neil, Shih-Chii Liu, and Tobi Delbruck. Ddd20 end-to-end event camera driving dataset: Fusing frames and events with deep learning for improved steering prediction. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2020.

[88] Xiaoling Huang. Image encryption algorithm using chaotic chebyshev generator. *Nonlinear Dynamics*, 67(4):2411–2417, 2012.

[89] Simone Undri Innocenti, Federico Becattini, Federico Pernici, and Alberto Del Bimbo. Temporal binary representation for event-based action recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10426–10432. IEEE, 2021.

[90] Haruyuki Iwama, Mayu Okumura, Yasushi Makihara, and Yasushi Yagi. The ou-isir gait database comprising the large population dataset and performance evaluation of gait recognition. *IEEE Transactions on Information Forensics and Security*, 7(5):1511–1521, 2012.

[91] Jam Jenkins and Carla Ellis. Using ground reaction forces from gait analysis: Body mass as a weak biometric. In *International conference on pervasive computing*, pages 251–267. Springer, 2007.

[92] Chaochuan Jia, Ting Yang, Chuanjiang Wang, Binghui Fan, and Fugui He. Encryption of 3d point cloud using chaotic cat mapping. *3D Research*, 10(1):4, 2019.

[93] Ning Jia, Victor Sanchez, and Chang-Tsun Li. Learning optimised representations for view-invariant gait recognition. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 774–780. IEEE, 2017.

[94] Zhuangyi Jiang, Pengfei Xia, Kai Huang, Walter Stechele, Guang Chen, Zhenshan Bing, and Alois Knoll. Mixed frame-/event-driven fast pedestrian detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8332–8338. IEEE, 2019.

[95] Xin Jin, Zhaoxing Wu, Chenggen Song, Chunwei Zhang, and Xiaodong Li. 3d point cloud encryption through chaotic mapping. In *Pacific Rim Conference on Multimedia*, pages 119–129. Springer, 2016.

[96] Alireza Jolfaei, Xin-Wen Wu, and Vallipuram Muthukkumarasamy. A 3d object encryption scheme which maintains dimensional and spatial stability. *IEEE Transactions on Information Forensics and Security*, 10(2):409–422, 2014.

[97] Amit Kale, AN Rajagopalan, Naresh Cuntoor, and Volker Kruger. Gait-based recognition of humans using continuous hmms. In *Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition*, pages 336–341. IEEE, 2002.

[98] Amit Kale, Aravind Sundaresan, AN Rajagopalan, Naresh P Cuntoor, Amit K Roy-Chowdhury, Volker Kruger, and Rama Chellappa. Identification of humans using gait. *IEEE Transactions on image processing*, 13(9):1163–1173, 2004.

[99] Alireza Khodamoradi and Ryan Kastner. O (n)-space spatiotemporal filter for reducing noise in neuromorphic vision sensors. *IEEE Transactions on Emerging Topics in Computing*, 2018.

[100] Hanme Kim, Ankur Handa, Ryad Benosman, Sio-Hoi Ieng, and Andrew J Davison. Simultaneous mosaicing and tracking with an event camera. *J. Solid State Circ*, 43:566–576, 2008.

[101] Hanme Kim, Stefan Leutenegger, and Andrew J Davison. Real-time 3d recon-

struction and 6-dof tracking with an event camera. In *European Conference on Computer Vision*, pages 349–364. Springer, 2016.

[102] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[103] Jurgen Kogler, Martin Humenberger, and Christoph Sulzbachner. Event-based stereo matching approaches for frameless address event stereo data. In *International Symposium on Visual Computing*, pages 674–685. Springer, 2011.

[104] Jürgen Kogler, Christoph Sulzbachner, Martin Humenberger, and Florian Eibensteiner. Address-event based stereo vision with bio-inspired silicon retina imagers. *Advances in theory and applications of stereo vision*, pages 165–188, 2011.

[105] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[106] Jorg Kramer. An on/off transient imager with event-driven, asynchronous read-out. In *2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No. 02CH37353)*, volume 2, pages II–II. IEEE, 2002.

[107] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[108] Beat Kueng, Elias Mueggler, Guillermo Gallego, and Davide Scaramuzza. Low-latency visual odometry using event-based feature tracks. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 16–23. IEEE, 2016.

[109] Worapan Kusakunniran, Qiang Wu, Jian Zhang, and Hongdong Li. Gait recognition under various viewing angles based on correlated motion regression. *IEEE transactions on circuits and systems for video technology*, 22(6):966–980, 2012.

[110] Xavier Lagorce, Cédric Meyer, Sio-Hoi Ieng, David Filliat, and Ryad Benosman. Asynchronous event-based multikernel algorithm for high-speed visual features tracking. *IEEE transactions on neural networks and learning systems*, 26(8):1710–1720, 2014.

[111] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E Shi, and Ryad B Benosman. Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1346–1359, 2017.

[112] Toby HW Lam, King Hong Cheung, and James NK Liu. Gait flow image: A silhouette-based gait representation for human identification. *Pattern recognition*, 44(4):973–987, 2011.

[113] Chankyu Lee, Adarsh Kumar Kosta, Alex Zihao Zhu, Kenneth Chaney, Kostas Daniilidis, and Kaushik Roy. Spike-flownet: event-based optical flow estimation with energy-efficient hybrid neural networks. In *European Conference on Computer Vision*, pages 366–382. Springer, 2020.

[114] Jun Haeng Lee, Tobi Delbruck, Michael Pfeiffer, Paul KJ Park, Chang-Woo Shin, Hyunsurk Ryu, and Byung Chang Kang. Real-time gesture interface based on event-driven processing from stereo silicon retinas. *IEEE transactions on neural networks and learning systems*, 25(12):2250–2263, 2014.

[115] KH Lee, H Woo, and T Suk. Point data reduction using 3d grids. *The International Journal of Advanced Manufacturing Technology*, 18(3):201–210, 2001.

[116] Lily Lee and W Eric L Grimson. Gait analysis for recognition and classification. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 155–162. IEEE, 2002.

[117] Gregor Lenz, Sio-Hoi Ieng, and Ryad Benosman. Event-based face detection and tracking in the blink of an eye. *arXiv preprint arXiv:1803.10106*, 2018.

[118] Na Li, Xinbo Zhao, and Chong Ma. A model-based gait recognition method based on gait graph convolutional networks and joints relationship pyramid mapping. *arXiv e-prints*, pages arXiv–2005, 2020.

[119] Shuangqun Li, Wu Liu, and Huadong Ma. Attentive spatial-temporal summary networks for feature learning in irregular gait recognition. *IEEE Transactions on Multimedia*, 21(9):2361–2375, 2019.

[120] Xiang Li, Yasushi Makihara, Chi Xu, Yasushi Yagi, and Mingwu Ren. Joint intensity transformer network for gait recognition robust against clothing and carrying status. *IEEE Transactions on Information Forensics and Security*, 14 (12):3102–3115, 2019.

[121] Xiang Li, Yasushi Makihara, Chi Xu, Yasushi Yagi, and Mingwu Ren. Gait recognition invariant to carried objects using alpha blending generative adversarial networks. *Pattern recognition*, 105:107376, 2020.

[122] Rijun Liao, Shiqi Yu, Weizhi An, and Yongzhen Huang. A model-based gait recognition method with body pose and human prior knowledge. *Pattern Recognition*, 98:107069, 2020.

[123] P Lichtsteiner. 64x64 event-driven logarithmic temporal derivative silicon retina. In *Program 2003 IEEE Workshop on CCD and AIS*, 2003.

[124] Patrick Lichtsteiner, Tobi Delbruck, and Jörg Kramer. Improved on/off temporally differentiating address-event imager. In *Proceedings of the 2004 11th IEEE International Conference on Electronics, Circuits and Systems, 2004. ICECS 2004.*, pages 211–214. IEEE, 2004.

[125] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128 $\times$ 128 120db 30mw asynchronous vision sensor that responds to relative intensity change. In *2006 IEEE International Solid State Circuits Conference-Digest of Technical Papers*, pages 2060–2069. IEEE, 2006.

[126] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128 $\times$ 128 120 db

15μs latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576, 2008.

[127] Beibei Lin, Shunli Zhang, and Feng Bao. Gait recognition with multiple-temporal-scale 3d convolutional neural network. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3054–3062, 2020.

[128] Dan Liu, Mao Ye, Xudong Li, Feng Zhang, and Lan Lin. Memory-based gait recognition. In *BMVC*, pages 1–12, 2016.

[129] Hongjie Liu, Christian Brandli, Chenghan Li, Shih-Chii Liu, and Tobi Delbruck. Design of a spatiotemporal correlation filter for event-based sensors. In *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, pages 722–725. IEEE, 2015.

[130] Jianyi Liu and Nanning Zheng. Gait history image: a novel temporal template for gait recognition. In *2007 IEEE international conference on multimedia and expo*, pages 663–666. IEEE, 2007.

[131] Shih-Chii Liu and Tobi Delbruck. Neuromorphic sensory systems. *Current opinion in neurobiology*, 20(3):288–295, 2010.

[132] Wenhao Liu, Kehui Sun, and Congxu Zhu. A fast image encryption algorithm based on chaotic map. *Optics and Lasers in Engineering*, 84:26–36, 2016.

[133] Wu Liu, Cheng Zhang, Huadong Ma, and Shuangqun Li. Learning efficient spatial-temporal gait features with deep learning for human identification. *Neuroinformatics*, 16(3):457–471, 2018.

[134] Yanxi Liu, Robert Collins, and Yanghai Tsin. Gait sequence analysis using frieze patterns. In *European Conference on Computer Vision*, pages 657–671. Springer, 2002.

[135] Zongyi Liu and Sudeep Sarkar. Simplest representation yet for gait recognition: Averaged silhouette. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, pages 211–214. IEEE, 2004.

[136] Patricio Loncomilla, Javier Ruiz-del Solar, and Luz Martínez. Object recognition using local invariant features for robotic applications: A survey. *Pattern Recognition*, 60:499–514, 2016.

[137] Chris Xiaoxuan Lu, Bowen Du, Xuan Kan, Hongkai Wen, Andrew Markham, and Niki Trigoni. Verinet: User verification on smartwatches via behavior biometrics. In *Proceedings of the First ACM Workshop on Mobile Crowdsensing Systems and Applications*, pages 68–73, 2017.

[138] Chris Xiaoxuan Lu, Bowen Du, Hongkai Wen, Sen Wang, Andrew Markham, Ivan Martinovic, Yiran Shen, and Niki Trigoni. Snoopy: Sniffing your smartwatch passwords via deep sequence learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):1–29, 2018.

[139] Chris Xiaoxuan Lu, Bowen Du, Peijun Zhao, Hongkai Wen, Yiran Shen, Andrew Markham, and Niki Trigoni. Deepauth: in-situ authentication for smartwatches via deeply learned behavioural biometrics. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*, pages 204–207, 2018.

[140] Chris Xiaoxuan Lu, Peijun Zhao, Bowen Du, Hongkai Wen, Andrew Markham, Stefano Rosa, and Niki Trigoni. Automatic face recognition adaptation via ambient wireless identifiers. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 377–378, 2018.

[141] Chris Xiaoxuan Lu, Xuan Kan, Bowen Du, Changhao Chen, Hongkai Wen, Andrew Markham, Niki Trigoni, and John Stankovic. Autonomous learning for face recognition in the wild via ambient wireless cues. In *The World Wide Web Conference*, pages 1175–1186, 2019.

[142] Jiwen Lu and Erhu Zhang. Gait recognition for human identification based on ica and fuzzy svm through multiple views fusion. *Pattern Recognition Letters*, 28(16):2401–2411, 2007.

[143] MA Mahowald and C Meader. The silicon retina. *Scientific American*, 264(5): 76–82, 1991.

[144] Davide Maltoni, Dario Maio, Anil K Jain, and Salil Prabhakar. *Handbook of fingerprint recognition*. Springer Science & Business Media, 2009.

[145] Jacques Manderscheid, Amos Sironi, Nicolas Bourdis, Davide Migliore, and Vincent Lepetit. Speed invariant time surface for learning to detect corner points with event-based cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10245–10254, 2019.

[146] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5419–5427, 2018.

[147] Iacopo Masi, Yue Wu, Tal Hassner, and Prem Natarajan. Deep face recognition: A survey. In *2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, pages 471–478. IEEE, 2018.

[148] Anton Mitrokhin, Chengxi Ye, Cornelia Fermüller, Yiannis Aloimonos, and Tobi Delbruck. Ev-imo: Motion segmentation dataset and learning pipeline for event cameras. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6105–6112. IEEE, 2019.

[149] Thor FR Mitskog and Richard Anthony Ralston. Camera blocker for a device with an integrated camera that uses a thin film organic polymer, November 29 2012. US Patent App. 13/477,485.

[150] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.

[151] Elias Mueggler, Basil Huber, and Davide Scaramuzza. Event-based, 6-dof pose tracking for high-speed maneuvers. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2761–2768. IEEE, 2014.

[152] Elias Mueggler, Nathan Baumli, Flavio Fontana, and Davide Scaramuzza. Towards evasive maneuvers with quadrotors using dynamic vision sensors. In *ECMR*, pages 1–8, 2015.

[153] Elias Mueggler, Chiara Bartolozzi, and Davide Scaramuzza. Fast event-based corner detection. 2017.

[154] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *The International Journal of Robotics Research*, 36(2):142–149, 2017.

[155] Gottfried Munda, Christian Reinbacher, and Thomas Pock. Real-time intensity-image reconstruction for event cameras using manifold regularisation. *International Journal of Computer Vision*, 126(12):1381–1393, 2018.

[156] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[157] Elaine M Newton, Latanya Sweeney, and Bradley Malin. Preserving privacy by de-identifying face images. *IEEE transactions on Knowledge and Data Engineering*, 17(2):232–243, 2005.

[158] Kien Nguyen, Clinton Fookes, Raghavender Jillela, Sridha Sridharan, and Arun Ross. Long range iris recognition: A survey. *Pattern Recognition*, 72:123–143, 2017.

[159] Sourabh A Niyogi, Edward H Adelson, et al. Analyzing and recognizing walking figures in xyt. In *CVPR*, volume 94, pages 469–474, 1994.

[160] MD Jan Nordin and Ali Saadoon. A survey of gait recognition based on skeleton model for human identification. *Research Journal of Applied Sciences, Engineering and Technology*, 12(7):756–763, 2016.

[161] Fernando Cladera Ojeda, Anthony Bisulco, Daniel Kepple, Volkan Isler, and Daniel D Lee. On-device event filtering with binary neural networks for pedes-

trian detection using neuromorphic vision sensors. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3084–3088. IEEE, 2020.

[162] Garrick Orchard, Ajinkya Jayawant, Gregory K Cohen, and Nitish Thakor. Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in neuroscience*, 9:437, 2015.

[163] Garrick Orchard, Cedric Meyer, Ralph Etienne-Cummings, Christoph Posch, Nitish Thakor, and Ryad Benosman. Hfirst: A temporal approach to object recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(10):2028–2040, 2015.

[164] Michael Otero. Application of a continuous wave radar for human gait recognition. In *Signal Processing, Sensor Fusion, and Target Recognition XIV*, volume 5809, pages 538–548. International Society for Optics and Photonics, 2005.

[165] Vandana Padala, Arindam Basu, and Garrick Orchard. A noise filtering algorithm for event-based asynchronous change detection image sensors on truenorth and its implementation on truenorth. *Frontiers in neuroscience*, 12: 118, 2018.

[166] Liyuan Pan, Miaomiao Liu, and Richard Hartley. Single image optical flow estimation with an event camera. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1669–1678. IEEE, 2020.

[167] Federico Paredes-Vallés and Guido CHE de Croon. Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy. *arXiv preprint arXiv:2009.08283*, 2020.

[168] Federico Paredes-Vallés, Kirk YW Scheper, and Guido CHE de Croon. Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception. *IEEE transactions on pattern analysis and machine intelligence*, 42(8):2051–2064, 2019.

[169] Narendra K Pareek, Vinod Patidar, and Krishan K Sud. Image encryption using chaotic logistic map. *Image and vision computing*, 24(9):926–934, 2006.

[170] Shwetak N Patel, Jay W Summet, and Khai N Truong. Blindspot: Creating capture-resistant spaces. In *Protecting Privacy in Video Surveillance*, pages 185–201. Springer, 2009.

[171] Ewa Piatkowska, Ahmed Belbachir, and Margrit Gelautz. Asynchronous stereo vision for event-driven dynamic stereo sensor using an adaptive cooperative approach. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 45–50, 2013.

[172] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds. *IEEE Journal of Solid-State Circuits*, 46(1): 259–275, 2011.

[173] Bibrat Ranjan Pradhan, Yeshwanth Bethi, Sathyaprakash Narayanan, Anirban Chakraborty, and Chetan Singh Thakur. N-har: A neuromorphic event-based human activity recognition system using memory surfaces. In *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2019.

[174] Lintian Qiao, Klara Nahrstedt, et al. A new algorithm for mpeg video encryption. In *Proc. of First International Conference on Imaging Science System and Technology*, pages 21–29, 1997.

[175] Bharath Ramesh and Hong Yang. Boosted kernelized correlation filters for event-based face detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 155–159, 2020.

[176] Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Real-time visualinertial odometry for event cameras using keyframe-based nonlinear optimization. In *British Machine Vis. Conf.(BMVC)*, volume 3, 2017.

[177] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. Events-to-video: Bringing modern computer vision to event cameras. In *Proceedings*

of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3857–3866, 2019.

[178] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[179] UZH Robotics and Perception Group. Event-based, 6-dof pose tracking for high-speed maneuvers using a dynamic vision sensor. `https://youtu.be/LauQ6LWTkxM`.

[180] Liu Rong, Zhou Jianzhong, Liu Ming, and Hou Xiangfeng. A wearable acceleration sensor system for gait recognition. In *2007 2nd IEEE Conference on Industrial Electronics and Applications*, pages 2654–2659. IEEE, 2007.

[181] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006.

[182] Bodo Rueckauer and Tobi Delbruck. Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor. *Frontiers in neuroscience*, 10:176, 2016.

[183] Samsung. Smartthings vision. `https://www.samsung.com/au/smartthings/camera/smartthings-vision-gp-u999gteeaac/`.

[184] Muhamad Risqi U Saputra, Andrew Markham, and Niki Trigoni. Visual slam and structure from motion in dynamic environments: A survey. *ACM Computing Surveys (CSUR)*, 51(2):1–36, 2018.

[185] Sudeep Sarkar, P Jonathon Phillips, Zongyi Liu, Isidro Robledo Vega, Patrick Grother, and Kevin W Bowyer. The humanid gait challenge problem: Data sets, performance, and analysis. *IEEE transactions on pattern analysis and machine intelligence*, 27(2):162–177, 2005.

[186] Stephan Schraml, Ahmed Nabil Belbachir, Nenad Milosevic, and Peter Schön. Dynamic stereo vision system for real-time tracking. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 1409–1412. IEEE, 2010.

[187] Alireza Sepas-Moghaddam and Ali Etemad. View-invariant gait recognition with attentive recurrent learning of partial representations. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(1):124–137, 2020.

[188] Alireza Sepas-Moghaddam, Saeed Ghorbani, Nikolaus F Troje, and Ali Etemad. Gait recognition using multi-scale partial representation transformation with capsules. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8045–8052. IEEE, 2021.

[189] Jagarlamudi Shashank, Palivela Kowshik, Kannan Srinathan, and CV Jawahar. Private content based image retrieval. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[190] Yiran Shen, Fengyuan Yang, Bowen Du, Weitao Xu, Chengwen Luo, and Hongkai Wen. Shake-n-shack: Enabling secure data exchange between smart wearables via handshakes. In *2018 IEEE international conference on pervasive computing and communications (PerCom)*, pages 1–10. IEEE, 2018.

[191] Yiran Shen, Bowen Du, Weitao Xu, Chengwen Luo, Bo Wei, Lizhen Cui, and Hongkai Wen. Securing cyber-physical social interactions on wrist-worn devices. *ACM Transactions on Sensor Networks (TOSN)*, 16(2):1–22, 2020.

[192] Kohei Shiraga, Yasushi Makihara, Daigo Muramatsu, Tomio Echigo, and Yasushi Yagi. Geinet: View-invariant gait recognition using a convolutional neural network. In *Biometrics (ICB), 2016 International Conference on*, pages 1–8. IEEE, 2016.

[193] Jamie D Shutler, Michael G Grant, Mark S Nixon, and John N Carter. On a large sequence-based human gait database. In *Applications and science in soft computing*, pages 339–346. Springer, 2004.

[194] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3693–3702, 2017.

[195] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1731–1740, 2018.

[196] Anna Sokolova and Anton Konushin. Human identification by gait from event-based camera. In *2019 16th International Conference on Machine Vision Applications (MVA)*, pages 1–6. IEEE, 2019.

[197] Bongki Son, Yunjae Suh, Sungho Kim, Heejae Jung, Jun-Seok Kim, Changwoo Shin, Keunju Park, Kyoobin Lee, Jinman Park, Jooyeon Woo, et al. 4.1 a 640× 480 dynamic vision sensor with a 9μm pixel and 300meps address-event representation. In *2017 IEEE International Solid-State Circuits Conference (ISSCC)*, pages 66–67. IEEE, 2017.

[198] Chunfeng Song, Yongzhen Huang, Yan Huang, Ning Jia, and Liang Wang. Gaitnet: An end-to-end network for gait based human identification. *Pattern recognition*, 96:106988, 2019.

[199] Shmuel Springer and Galit Yogev Seligmann. Validity of the kinect for gait assessment: A focused review. *Sensors*, 16(2):194, 2016.

[200] Rasmus Stagsted, Antonio Vitale, Jonas Binz, Leon Bonde Larsen, Yulia Sandamirskaya, et al. Towards neuromorphic control: A spiking neural network based pid controller for uav. RSS, 2020.

[201] Timo Stoffregen, Guillermo Gallego, Tom Drummond, Lindsay Kleeman, and Davide Scaramuzza. Event-based motion segmentation by motion compensation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7244–7253, 2019.

[202] Han Su and Fenggang Huang. Gait recognition using principal curves and neural networks. In *International Symposium on Neural Networks*, pages 238–243. Springer, 2006.

[203] Jingran Su, Yang Zhao, and Xuelong Li. Deep metric learning based on center-ranked loss for gait recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4077–4081. IEEE, 2020.

[204] Subramania Sudharsanan. Shared key encryption of jpeg color images. *IEEE Transactions on Consumer Electronics*, 51(4):1204–1211, 2005.

[205] Jaakko Suutala and Juha Roning. Combining classifiers with different footstep feature sets and multiple samples for person identification. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 5, pages v–357. IEEE, 2005.

[206] Jaakko Suutala and Juha Röning. Methods for person identification on a pressure-sensitive floor: Experiments with multiple classifiers and reject option. *Information Fusion*, 9(1):21–40, 2008.

[207] Noriko Takemura, Yasushi Makihara, Daigo Muramatsu, Tomio Echigo, and Yasushi Yagi. On input/output architectures for convolutional neural network-based cross-view gait recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(9):2708–2719, 2017.

[208] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. Visual slam algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):1–11, 2017.

[209] Daoliang Tan, Kaiqi Huang, Shiqi Yu, and Tieniu Tan. Efficient night gait recognition based on template matching. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 1000–1003. IEEE, 2006.

[210] Rawesak Tanawongsuwan and Aaron Bobick. Gait recognition from time-normalized joint-angle trajectories in the walking plane. In *Proceedings of*

the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, volume 2, pages II–II. IEEE, 2001.

[211] Dacheng Tao, Xuelong Li, Xindong Wu, and Stephen J Maybank. General tensor discriminant analysis and gabor features for gait recognition. *IEEE transactions on pattern analysis and machine intelligence*, 29(10), 2007.

[212] Ran Tao, Xiang-Yi Meng, and Yue Wang. Image encryption with multiorders of fractional fourier transforms. *IEEE transactions on Information Forensics and Security*, 5(4):734–738, 2010.

[213] Gemma Taverni, Diederik Paul Moeys, Chenghan Li, Celso Cavaco, Vasyl Motsnyi, David San Segundo Bello, and Tobi Delbruck. Front and back illuminated dynamic and active pixel vision sensors comparison. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 65(5):677–681, 2018.

[214] Suibing Tong, Hefei Ling, Yuzhuo Fu, and Dan Wang. Cross-view gait identification with embedded learning. In *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, pages 385–392, 2017.

[215] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[216] Valentina Vasco, Arren Glover, and Chiara Bartolozzi. Fast event-based harris corner detection exploiting the advantages of event-driven cameras. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4144–4149. IEEE, 2016.

[217] Valentina Vasco, Arren Glover, Elias Mueggler, Davide Scaramuzza, Lorenzo Natale, and Chiara Bartolozzi. Independent motion detection with event-driven cameras. In *2017 18th International Conference on Advanced Robotics (ICAR)*, pages 530–536. IEEE, 2017.

[218] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones,

Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[219] Antoni Rosinol Vidal, Henri Rebecq, Timo Horstschaefer, and Davide Scaramuzza. Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios. *IEEE Robotics and Automation Letters*, 3(2):994–1001, 2018.

[220] Changsheng Wan, Li Wang, and Vir V Phoha. A survey on gait recognition. *ACM Computing Surveys (CSUR)*, 51(5):1–35, 2018.

[221] Liang Wang, Tieniu Tan, Huazhong Ning, and Weiming Hu. Silhouette analysis-based gait recognition for human identification. *IEEE transactions on pattern analysis and machine intelligence*, 25(12):1505–1518, 2003.

[222] Liang Wang, Huazhong Ning, Tieniu Tan, and Weiming Hu. Fusion of static and dynamic body biometrics for gait recognition. *IEEE Transactions on circuits and systems for video technology*, 14(2):149–158, 2004.

[223] Lin Wang, Yo-Sung Ho, Kuk-Jin Yoon, et al. Event-based high dynamic range image and very high frame rate video generation using conditional generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10081–10090, 2019.

[224] Lin Wang, Tae-Kyun Kim, and Kuk-Jin Yoon. Eventsr: From asynchronous events to image reconstruction, restoration, and super-resolution via end-to-end adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8315–8325, 2020.

[225] Lin Wang, Yujeong Chae, Sung-Hoon Yoon, Tae-Kyun Kim, and Kuk-Jin Yoon. Evdistill: Asynchronous events to end-task learning via bidirectional reconstruction-guided cross-modal knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 608–619, 2021.

[226] Qinyi Wang, Yexin Zhang, Junsong Yuan, and Yilong Lu. Space-time event clouds for gesture recognition: From rgb cameras to event cameras. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1826–1835. IEEE, 2019.

[227] Yanxiang Wang, Bowen Du, Yiran Shen, Kai Wu, Guangrong Zhao, Jianguo Sun, and Hongkai Wen. Ev-gait: Event-based robust gait recognition using dynamic vision sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6358–6367, 2019.

[228] Yanxiang Wang, Xian Zhang, Yiran Shen, Bowen Du, Guangrong Zhao, Cui Lizhen, and Hongkai Wen. Event-stream representation for human gaits identification using deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[229] Yanyun Wang, Chunfeng Song, Yan Huang, Zhenyu Wang, and Liang Wang. Learning view invariant gait features with two-stream gan. *Neurocomputing*, 339:245–254, 2019.

[230] Zhihao Wang, Jian Chen, and Steven CH Hoi. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[231] Susie J Wee and John G Apostolopoulos. Secure scalable video streaming for wireless networks. In *IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS SPEECH AND SIGNAL PROCESSING*, volume 4, pages IV–2049. IEEE; 1999, 2001.

[232] David Weikersdorfer and Jörg Conradt. Event-based particle filtering for robot self-localization. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 866–870. IEEE, 2012.

[233] Hongkai Wen, Ronald Clark, Sen Wang, Xiaoxuan Lu, Bowen Du, Wen Hu, and Niki Trigoni. Efficient indoor positioning with visual experiences via

lifelong learning. *IEEE Transactions on Mobile Computing*, 18(4):814–829, 2018.

[234] Thomas Wolf, Mohammadreza Babaee, and Gerhard Rigoll. Multi-view gait recognition using 3d convolutional neural networks. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 4165–4169. IEEE, 2016.

[235] Chung-Ping Wu and C-C Jay Kuo. Efficient multimedia encryption via entropy codec design. In *Security and Watermarking of Multimedia Contents III*, volume 4314, pages 128–138. International Society for Optics and Photonics, 2001.

[236] Zifeng Wu, Yongzhen Huang, Liang Wang, Xiaogang Wang, and Tieniu Tan. A comprehensive study on cross-view gait based human identification with deep cnns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

[237] Zifeng Wu, Yongzhen Huang, Liang Wang, Xiaogang Wang, and Tieniu Tan. A comprehensive study on cross-view gait based human identification with deep cnns. *IEEE transactions on pattern analysis and machine intelligence*, 39(2): 209–226, 2016.

[238] Zhaopeng Xu, Wei Lu, Qin Zhang, Yuileong Yeung, and Xin Chen. Gait recognition based on capsule network. *Journal of Visual Communication and Image Representation*, 59:159–167, 2019.

[239] ChewYean Yam, Mark S Nixon, and John N Carter. Automated person recognition by walking and running via model-based approaches. *Pattern recognition*, 37(5):1057–1072, 2004.

[240] Chao Yan, Bailing Zhang, and Frans Coenen. Multi-attributes gait identification by convolutional neural networks. In *2015 8th International Congress on Image and Signal Processing (CISP)*, pages 642–647. IEEE, 2015.

[241] Jang-Hee Yoo, Doosung Hwang, Ki-Young Moon, and Mark S Nixon. Automated human recognition by gait using neural network. In *2008 First Workshops on Image Processing Theory, Tools and Applications*, pages 1–6. IEEE, 2008.

[242] Lei Yu, Wen Yang, et al. Event-based high frame-rate video reconstruction with a novel cycle-event network. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 86–90. IEEE, 2020.

[243] Shiqi Yu, Daoliang Tan, and Tieniu Tan. A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 4, pages 441–444. IEEE, 2006.

[244] Shiqi Yu, Haifeng Chen, Edel B Garcia Reyes, and Norman Poh. Gaitgan: Invariant gait feature extraction using generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 30–37, 2017.

[245] Wenzhen Yuan and Srikumar Ramalingam. Fast localization and tracking using event sensors. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4564–4571. IEEE, 2016.

[246] Cheng Zhang, Wu Liu, Huadong Ma, and Huiyuan Fu. Siamese neural network based gait recognition for human identification. In *2016 ieee international conference on acoustics, speech and signal processing (ICASSP)*, pages 2832–2836. IEEE, 2016.

[247] Erhu Zhang, Yongwei Zhao, and Wei Xiong. Active energy image plus 2dlpp for gait recognition. *Signal Processing*, 90(7):2295–2302, 2010.

[248] Kaihao Zhang, Wenhan Luo, Lin Ma, Wei Liu, and Hongdong Li. Learning joint gait representation via quintuplet loss minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4700–4709, 2019.

[249] Liang Zhang, Guangming Zhu, Peiyi Shen, Juan Song, Syed Afaq Shah, and Mohammed Bennamoun. Learning spatiotemporal features using 3dcnn and convolutional lstm for gesture recognition. In *Proceedings of the IEEE In-*

ternational Conference on Computer Vision Workshops, pages 3120–3128, 2017.

[250] Yuqi Zhang, Yongzhen Huang, Liang Wang, and Shiqi Yu. A comprehensive study on gait biometrics using a joint cnn-based method. *Pattern Recognition*, 93:228–236, 2019.

[251] Yuqi Zhang, Yongzhen Huang, Shiqi Yu, and Liang Wang. Cross-view gait recognition by discriminative feature learning. *IEEE Transactions on Image Processing*, 29:1001–1015, 2019.

[252] Ziyuan Zhang, Luan Tran, Xi Yin, Yousef Atoum, Xiaoming Liu, Jian Wan, and Nanxin Wang. Gait recognition via disentangled representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4710–4719, 2019.

[253] Ziyuan Zhang, Luan Tran, Feng Liu, and Xiaoming Liu. On learning disentangled representations for gait recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[254] Nanrun Zhou, Xingbin Liu, Ye Zhang, and Yixian Yang. Image encryption scheme based on fractional mellin transform and phase retrieval technique in fractional fourier domain. *Optics & Laser Technology*, 47:341–346, 2013.

[255] Alex Zihao Zhu and Liangzhe Yuan. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. In *Robotics: Science and Systems*, 2018.

[256] Alex Zihao Zhu, Yibo Chen, and Kostas Daniilidis. Realtime time synchronized event-based stereo. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 433–447, 2018.

[257] Alex Zihao Zhu, Dinesh Thakur, Tolga Özaslan, Bernd Pfrommer, Vijay Kumar, and Kostas Daniilidis. The multivehicle stereo event camera dataset: An event camera dataset for 3d perception. *IEEE Robotics and Automation Letters*, 3(3): 2032–2039, 2018.

[258] Alex Zihao Zhu, Ziyun Wang, Kaung Khant, and Kostas Daniilidis. Eventgan: Leveraging large scale image datasets for event cameras. In *2021 IEEE International Conference on Computational Photography (ICCP)*, pages 1–11. IEEE, 2021.