Asynchronous Optical Flow and Egomotion Estimation from Address Events Sensors

by

Charbel Azzi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2022

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Medhat Moussa Professor, School of Engineering University of Guelph
Supervisor(s):	Eihab Abdel-Rahman Professor, Systems Design Engineering University of Waterloo
	Adel Fakih Adjunct Professor, Systems Design Engineering University of Waterloo
Internal Member:	Paul Fieguth Associate Dean, Systems Design Engineering University of Waterloo
Other Member(s):	Alexander Wong Professor, Systems Design Engineering University of Waterloo

Internal-External Member: William Melek
Professor, Mechanical and Mechatronics Engineering
University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Motion estimation is considered essential for many applications such as robotics, automation, and augmented reality to name a few. All cheap and low cost sensors which are commonly used for motion estimation have many shortcomings. Recently, event cameras are a new stream in imaging sensor technology characterized by low latency, high dynamic range, low power and high resilience to motion blur. These advantages allow them to have the potential to fill some of the gaps of other low cost motion sensors, offering alternatives to motion estimation that are worth exploring.

All current event-based approaches estimate motion by considering that events in a neighborhood encode the local structure of the imaged scene, then track the evolution of this structure over time which is problematic since events are only an approximation of the local structure that can be very sparse in some cases. In this thesis, we tackle the problem in a fundamentally different way by considering that events generated by the motion of the same scene point relative to the camera constitute an *event track*. We show that consistency with a single camera motion is sufficient for correct data association of events and their previous firings along event tracks resulting in more accurate and robust motion estimation.

Towards that, we present new voting based solutions which consider all potential data association candidates that are consistent with a single camera motion for candidates evaluation by handling each event individually without assuming any relationship to its neighbors beyond the camera motion. We first exploit this in a particle filtering framework for the simple case of a camera undergoing a planar motion, and show that our approach can yield motion estimates that are an order of magnitude more accurate than optical flow based approaches. Furthermore, we show that the consensus based approach can be extended to work even in the case of arbitrary camera motion and unknown scene depth. Our general motion framework significantly outperforms other approaches in terms of accuracy and robustness.

Acknowledgements

I am indebted to numerous people without whom my PhD studies would have been a much harder and less fruitful experience, and to only few of them, I can give a particular mention here.

I had the privilege to be supervised by Dr.Eihab Abdel-Rahman and Dr.Adel Fakih who offered me their assistance and unequivocal support in all possible fashions. I would like to express my sincere gratitude to them for that and for their valuable guidance, critical advice,and patience and for being great academic role models.

Many thanks are extended to the members of my committee, professors, Medhat Moussa, William Melek,, Paul Fieguth, and Alexander Wong for reading my thesis and for providing helpful and insightful comments.

I convey special acknowledgement to two very special families who became my family here. They made my stay in Waterloo enjoyable and made me feel at home. I will always remember the great times we spent together, the delicious meals they generously prepared, and how you stood by my side in the toughest moments of my life. I will be always be indebted for you.

To Bank Audi who funded part of my expenses throughout this degree I would like to thank you a lot for believing in me. I would not have accomplished this work without your full support. I will always owe you for the opportunities you have given me.

Words fail me to express my heartfelt gratitude for my parents who, through my childhood and study career, provided me with all their love and encouragement and worked hard to secure me an excellent education. I regret that I would never be able to pay them back for all the years I spent away doing a Masters degree then a PhD degree.

Last but not least, I owe my loving thanks to my fiancee Jennifer whose presence, love, and never-ending support were a constant source of motivation and inspiration that kept me going through my PhD journey.

Dedication

To my parents, fiancee, and a special best friend for their sacrifices, support, and belief in me.

Table of Contents

List of Figures

List of Tables

Acronyms

AEE Average Endpoint Error. 54

AES Address Event Sensors. 1

BCE Brightness Constancy Equation. 13

EV-OF Event-based Optical Flow. 16

HDR High Dynamic Range. 54, 77

IMU Inertial Measurement Unit. 77

LK Lucas and Kanade. 61

OF Optical Flow. 10

Chapter 1

Introduction

1.1 Motivation

Motion estimation is considered indispensable for various tasks including robotics, automation, surveillance, and augmented reality to name a few. Cheap and low cost sensors are commonly used for motion estimation, from Inertial Measurement Unit (IMU), to encoders, to cameras, to optical trackers, etc. All low cost motion sensors are not perfect and have many shortcomings when it comes to estimating motion such as accuracy, robustness, failure mode, power consumption, need for calibration, bias, environmental conditions effect, operational temperature, and so on.

Recently, there has been notable progress in imaging sensor technology, which can offer alternative solutions to motion estimation. In particular, neuromorphic imaging devices, called event cameras (also known as Address Event Sensors (AES)) [22, 94, 141], are asynchronous, cheap, and low cost sensors that mimic the human visual system by responding to changes in illumination based on the scene's dynamics. They do not record image frames, but a stream of asynchronous events at microsecond resolution, each of which is immediately generated when a given pixel detects a change in log intensity. This enables the event cameras to *see the motion* in the scene, which makes

them attractive to motion estimation. Their main advantages come from their low latency, high dynamic range, and resiliency to motion blur, while consuming small amount of power.

The advantages of event cameras allow them to have the potential to fill some of the gaps of other low cost motion sensors, offering alternatives to motion estimation that are worth exploring. However, current event-based motion estimation approaches still have ample of room for improvements. In this thesis, we believe that we can exploit event cameras in a different way to achieve better motion estimation.

1.2 Premise

Event cameras have characteristics that, in some situations, would make them a better alternative for motion estimation than traditional frame-based cameras.

1.2.1 Low Latency

Event cameras measure brightness changes with a very high temporal resolution, which are reported with a very low latency in the order of microseconds. This is four orders of magnitude higher compared to the 30-60Hz fixed rate achieved by traditional cameras of comparable price and power consumption. Therefore, event cameras can perceive the scene dynamics with much finer temporal details, which allows them to capture very fast motions that can be used for high speed tracking applications for instance. In order to achieve similar adaption to situations with rapid motion, traditional cameras need to capture thousands of frames every second, which results in a large amount of data to be transmitted and processed, where such cameras will no longer fall in the category of cheap and low cost sensors.

1.2.2 High Dynamic Range

In frame-based cameras, the dynamic range is typically around 60dB which leads to the issue of over or under exposure, where correct exposure for one part of an image can be considered too low or high for another. This leads to loss of information in situations with strong lighting changes. This is common in robotics for instance, where a robot moving in the sun, or transitioning between low and high lightning scenes. In such situations, event cameras can be a better alternative since every pixel is independent, and track of the log intensity changes (smaller change in the absolute magnitude), they can respond better to wide range of lighting, by reaching very high dynamic ranges of the order of 140db. Frame-based cameras can be tuned to reach higher dynamic ranges by using different exposure time but at the expense of an increased latency, which would result in blurred images in dark scenes for example.

1.2.3 Resiliency to Motion Blur

Traditional cameras have a global shutter speed which exposes all their pixels simultaneously by collecting light during a given exposure time. This can result in motion blur in situations with fast motion, or when large intensity difference are present (bright or dark scenes). In these situations, event cameras are more robust since all their pixels are independent, eliminating the need for a global exposure time. The exposure of traditional cameras can be increased by using a faster shutter speed to reduce motion blur, but there is always a limit on the shutter speed increase. Additionally, this results in an increase in the latency, and if the scene is not bright enough, higher shutter speeds can result in underexposed images that are bad for motion estimation.

1.2.4 Low Power

Event cameras have significantly lower power consumption than traditional cameras. Since only changes in brightness are streamed in their output, they only transmit non-redundant data instead of full frames, thus, reducing the power necessary for acquisition and transmission. This can be beneficial for situations with fast and robust motion where resources are constrained such as embedded devices or some robotics applications.

1.3 Objectives & Contributions

A common factor in all previous event-based motion estimation approaches is the view that events in a neighborhood encode the local structure of the imaged scene. Then, motion estimation is performed by tracking the evolution of this structure over time. The problem with this view is that events are only an approximation of the local structure that can be very sparse in some cases.

This thesis aims at taking a radically different view of the problem: events generated by the motion of the same scene point relative to the camera constitute an *event track*. These event tracks provide more accurate constraints on the camera motion than the evolution of the local structure as they are not subjected to any approximation other than image quantization and brightness constancy. The challenge however is in matching events to their previous firing locations.

We make the hypothesis that, in the case of a dominant motion between the camera and the scene, consistency with the dominant motion is sufficient for correct data association of events and their previous firings along event tracks.

This thesis explores using voting based approaches with an exhaustive search over all possible data associate candidates to select the ones that are consistent with a single camera motion. We show that this can successfully

identify the correct motion in almost all situations and we characterize cases where it could fail. We exploit this in a particle filtering framework that, for the simple case of a camera undergoing a single translation parallel to a planar wall, can yield motion estimates that are an order of magnitude more accurate than optical flow based approaches.

Furthermore, we show that the consensus based approach can be extended to work even in the case of an arbitrary camera motion and an arbitrary scene depth in which a single data association can vote for multiple motions. We showcase this in a particle filter based motion estimation system that significantly outperforms other approaches in terms of accuracy and robustness.

Chapter 2

Event Cameras

In this chapter, we present a brief overview on how event cameras work. For a deeper understanding about these sensors, we refer the reader to the recent survey [59].

2.1 Overview

Event cameras[94, 22, 141], also known as Address Event Sensors (AES), are bio-inspired neuromorphic sensors. They closely mimic the human eye where the human ganglion cell fires independently as soon as a light is detected by they eye's retina in response to a brightness changes. Similarly, the AES fires events at individual pixels as soon as a brightness change is detected. In contrast, traditional or frame-based cameras generate images resembling the final output we see with our eyes bypassing many layers of the human eye way of operation.

Table 2.1 compares an expensive high frequency camera (Photron), traditional camera (Bluefox), and an event camera (DVS). Contrary to a regular camera which output frames at fixed time rate of 90 Hz, the event camera generate asynchronous events at low latency of 1 MHz or 1μs. It also enjoys a much lower storage transmission bandwidth and lower power consumption

	Photron Fastcam SA5	Matrix Vision Bluefox	DVS
Max fps or measurement rate	1MHz	90 Hz	**1MHz**
Resolution at max fps	64x16 pixels	752x480 pixels	**128x128 pixels**
Bits per pixels	12 bits	8-10	**1 bits**
Weight	6.2 Kg	30 g	**30 g**
Active cooling	yes	No cooling	**No cooling**
Data rate	1.5 GB/s	32MB/s	**~200KB/s on average**
Power consumption	150 W + Ilighting	1.4 W	**20 mW**
Dynamic range	n.a.	60 dB	**120 dB**

Table 2.1: Comparison between the main types of cameras.

than the other two cameras. Using expensive motion sensors such as high frequency cameras can achieve similar latency as the event cameras, however, it comes at more then 10 folds the price and requires a very high power consumption, data transmission and memory requirements, and processing time.

Figure 2.1 visualizes the output of AES versus the output of standard frame-based cameras, when looking at a black dot on a rotating disk. The asynchronous output is clearly visible for the AES while the disk rotates, whereas we can clearly see that when the disk stops rotating, no events are generated from the AES while the regular camera keeps sending redundant images. The main disadvantages of the AES are their low spatial resolution and lack of intensity levels. However, the intensity levels could be recovered by spatial and temporal processing.

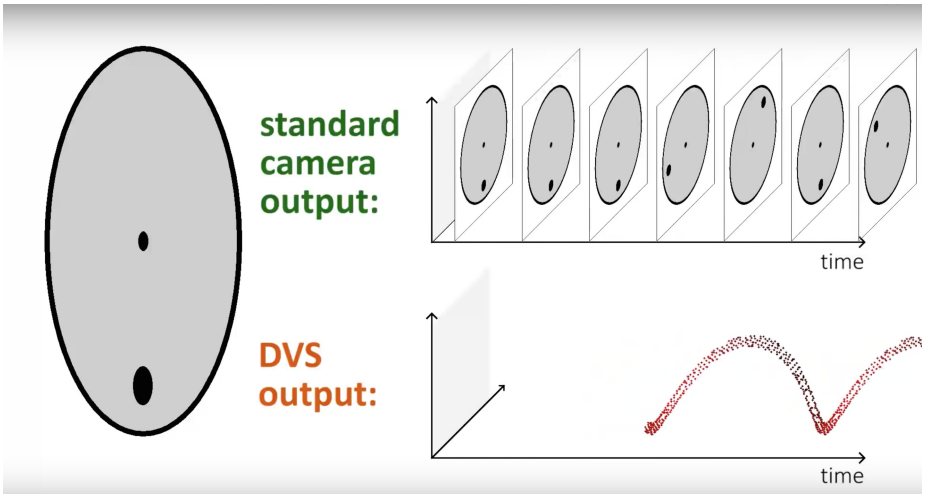


Figure 2.1: Output visualization from AES compared to standard frame-based cameras when looking at a black dot on a rotating disk [119].

2.2 Principle of Operation

The asynchronous transmission of an event at the time it occurs is done using digital circuitry. The simplified diagram of the circuitry illustrated in the left panel of Fig. 2.2 shows that an event camera has independent pixels that triggers whenever their 'log intensity (photocurrent)', referred to as 'brightness' change. Each pixel memorizes the log intensity value each time it fires an event. It continuously monitors the brightness level of the current pixel location $\mathbf{x}$ at time t until it exceeds a certain threshold δ_e with respect to the memorized value at a previously referenced time t_{ref}:

$$\Delta I(\mathbf{x}, t) = I(\mathbf{x}, t) - I(\mathbf{x}, t_{ref}) \geq \delta_e \tag{2.1}$$

at which the camera fires an event e transmitted by the chip in a quadruple data format $e = \langle \mathbf{x}, t, p \rangle$, where $\mathbf{x} = (x, y)$ are the 2D pixel coordinates, t is the corresponding microsecond timestamp, and $p \in \{1, -1\}$ corresponds to the polarity of the brightness change (1 denotes an increasing brightness ('ON') from dark to light and -1 denotes a decreasing brightness ('OFF')

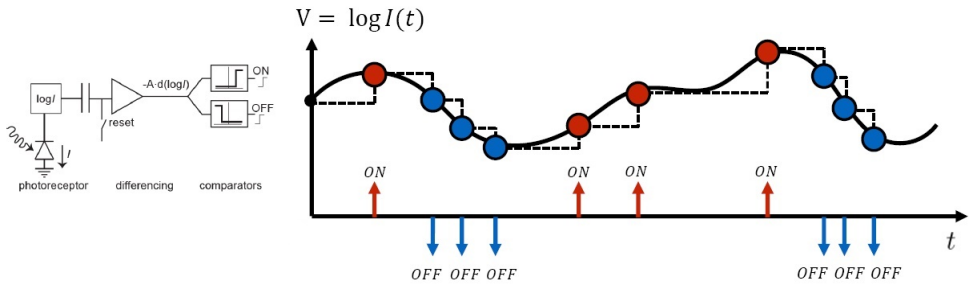


Figure 2.2: Simplified circuit diagram and operation of an AES [94].

from light to dark). This event firing is illustrated in the right panel of Fig. 2.2. All events are outputted from the camera's pixel array via a shared digital output bus [175, 102]. This bus can in some situations cause some bandwidth limitations which perturbs the the times at which events are sent (See Appendix A).

2.3 Conclusion

Event cameras are cheap and low cost motion sensors with precise timing, high sampling frequency, high dynamic range and low power consumption. These characteristics offer the potential to achieve accurate and robust motion estimation.

Chapter 3

Problem Formulation

This thesis aims to estimate the relative motion (output) between a scene and an event camera (AES) given events (input) captured by this camera.

3.1 Preliminaries

Throughout this thesis the following notation will be adopted:

- The pixel location is denoted:

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$$

  and its velocity is denoted:

$$\boldsymbol{U} = \begin{bmatrix} u \\ v \end{bmatrix}$$

  where u and v are its horizontal and vertical components also known as the Optical Flow (OF).

- Let $I(\mathbf{x}, t)$ denote the log intensity 'brightness' at position $\mathbf{x}$ and time t.

- $e(\mathbf{x}, t, p)$ denotes an event at position $\mathbf{x}$ and time t with a polarity defined by

$$\begin{cases} p = 1; & I(\mathbf{x}, t) - I(\mathbf{x}, t - \delta t) > \delta_e \\ p = -1; & I(\mathbf{x}, t) - I(\mathbf{x}, t - \delta t) < -\delta_e, \end{cases}$$

  where δt is the sampling period and δ_e denote the brightness threshold required to trigger an event.

- The function $\Pi_{i,j}$:

$$\Pi_{i,j} = \mathbf{x} + (t_j - t_i)\mathbf{U} \tag{3.1}$$

  projects a pixel from time t_i to time t_j.

3.2 Event Formation

The motion of an event camera relative to a static 3D scene (known as egomotion) induces a 2D velocity field of the intensities in the image plane. Each point $\mathbf{P} = (X, Y, Z)$ in the scene moves along a 3D path, with a relative velocity with respect the camera. The egomotion of the event camera is composed of the translational velocity

$$\mathbf{T} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \in \mathbb{R}^3$$

and the angular velocity

$$\mathbf{\Omega} = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \in SO(3)$$

The equations governing the velocity field $\mathbf{U} = (u, v)$ are derived by projecting the 3D relative velocities of the scene points onto the image

plane [103]. The equations of the 2D velocity field are derived using the pin hole camera projection model as:

$$u = f\omega_y - y\omega_z + \frac{x^2}{f}\omega_y - \frac{xy}{f}\omega_x + f\frac{T_x}{Z} - x\frac{T_z}{Z} \tag{3.2}$$

$$v = x\omega_z - f\omega_x - \frac{y^2}{f}\omega_x + \frac{xy}{f}\omega_y + f\frac{T_y}{Z} - y\frac{T_z}{Z} \tag{3.3}$$

where f is the focal length of the camera and Z is the depth of the scene (each pixel having its own depth).

The camera records an event when the brightness of a point in the 3D scene projected onto the image plane changes from I_0 to I_1 such the constraint (2.1) is satisfied. The egomotion of the camera causes instantaneous displacements of a set of points $\{P_k \in \mathbb{R}^3\}_{i=1}^M$ and their projections $\{p_k \in \mathbb{R}^2\}_{i=1}^M$ onto the event camera's array. This results in events firing due to *motion*. Additionally, changes in *illumination* conditions also lead to events firing at pixels where brightness changes such that $dI > \delta e$. This results in a *stream of events* $\mathcal{E} = \{e_i\}_{i=1}^N$ that is generated at different timestamps $t_1, ..., t_N$, due to motion and illumination.

3.3 Event-based Motion Estimation

Problem. *Given an event stream $\mathcal{E}$ generated by the projection of motion and illumination changes onto the event camera's plane, estimate the relative motion between the scene and the camera over time.*

The velocity field equations (3.2) and (3.3) relate the optical flow $\mathbf{U}$ to the relative motion of the event camera $(\mathbf{T}, \mathbf{\Omega})$, the pixel depth Z, and its location $\mathbf{x}$:

$$\mathbf{U} = \mathbf{f}(\mathbf{T}, \mathbf{\Omega}, Z, \mathbf{x}) \tag{3.4}$$

When the camera moves, a stream of events $\mathcal{E}$ is generated. To solve (3.4) we need to establish a relationship between those measurements and the camera

motion. An event e firing due motion is related to a previously fired event e_p at location (x_p, y_p) and time $t - m\,\delta t$ by

$$\mathbf{h}(\mathbf{U}, e, e_p) = 0$$

On the other hand, spurious events, triggered by uncorrelated illumination changes, have no priors. We can define a function for any event as:

$$\begin{cases} \mathbf{g}(\mathbf{U}, e, e_p, dI) = 0; & \text{if } dI < \delta_e \\ \mathbf{f}(e, dI) = 0; & \text{otherwise,} \end{cases} \tag{3.5}$$

where g associates the stream of events and their priors, note that if $dI = 0$ then $g = h$. The function f has no bearing on motion estimation.

The event-based motion estimation problem is mathematically formulated by Eq. (3.4) and (3.5). The optical flow $\mathbf{U}$ is estimated through Eq. (3.5) by comparing a current event e to its prior e_p. It is then used to estimate the camera motion $(\mathbf{T}, \mathbf{\Omega})$ through Eq. (3.4).

3.3.1 Illumination Change dI

Estimating dI is virtually impossible without making assumptions about the light sources and surface properties. It is frequently circumvented by adopting the Brightness Constancy Equation (BCE) which sets: $dI = 0$.

3.3.2 Motion Model

Additionally, we do not have a clear model of the function g. A common assumption adopted is a constant velocity model as the motion function $\mathbf{g}$ and set

$$\mathbf{x_p} = \mathbf{x} + (t_j - t_i)\mathbf{U}$$

to project an event from $(\mathbf{x}, t_i)$ to $(\mathbf{x_p}, t_j)$. Assuming the BCE and the constant velocity model, Eq (3.5) reduces to:

$$\mathbf{g} = \mathbf{h} = 0 \implies e_p = \Pi(\mathbf{U}, e) \tag{3.6}$$

3.3.3 Data Association Problem

Solving Eq. (3.6) requires solving for the *data association* between an event e and its previous firing e_p. Existing solutions of the *data association* problem assume that events approximate the local image structure around the current event. This allows them to employ standard image matching or tracking techniques. However, events in many situations provide only a sparse approximation of the local structure that might not be enough for matching or tracking.

3.3.4 Undetermined Problem

The reduced problem in Eqs. (3.4) and (3.6) is ill-posed (undetermined) as it has more unknowns than equations. Current solutions overcome this challenge by assuming smoothness with neighboring events having similar velocities. On the other hand, event cameras are meant to be used in sparse natural scenes where events are generated along object boundaries or edges. In these situations, the spatial smoothness assumption can be easily violated, thereby deteriorating the accuracy of the motion estimation.

3.4 Problem Statement

This thesis hypothesizes that avoiding the assumptions discussed in 3.3.3 and 3.3.4 will result in more robust and accurate motion estimation. Towards that, it proposes solving Eqs. (3.4) and (3.6) by handling each event individually without assuming any relationship to its neighbors beyond the common camera motion. To resolve the data association and the ill-posedness

problems, we explore new voting-based solutions that rely on considering all potential data association candidates that are consistent with a single camera motion for candidates evaluation.

Chapter 4

Literature Review

In this chapter, we review the related work to event-based motion estimation. The first part reviews Event-based Optical Flow (EV-OF) to compute image velocities from a stream of events, then egomotion estimation using event cameras. The second part focuses on reviewing the objective functions functions to highlight their effect in tackling the data association and ill-posedness challenges for event-based motion estimation. Further, we present a review on how filtering techniques are used in event-based motion estimation. Finally, we present a summary of the major limitations for motion estimation.

4.1 Event-based Optical Flow

Optical flow is the problem of computing the velocity or motion of objects on the image plane. event cameras are attractive for OF estimation since events represent edges in natural scenes where OF estimation is less ambiguous, and due to their precise timing which allow to measure high speed flow. The main optical flow approaches can be divided into three categories:

- Classical approaches that are based directly adopting frame-based algorithms.

- Variational approaches (the dominant stream) that consider a local smooth distribution of events in x-y-t space.
- Deep learning approaches that rely on recent successful machine-learning approaches from frame-based optical flow.

4.1.1 Classical Approaches

Early motion estimation from event cameras followed the classical approaches used in frame-based techniques. Similarly, it uses the gradient constraint equation based on the brightness constancy assumption. In event-based vision there are no intensity information, therefore a formulation of intensity from event cameras was the major focus of these approaches.

Benosman et al. [19] introduced the first BCE-based approach. This method adapts the Lucas and Kanade local smoothness assumption to estimate the temporal derivative of the brightness over the local image structure. Brosch et al. [23] added a second order term to the temporal gradient estimate in Eq. (4.1) and (4.2) , to make the derivative estimation more stable. Although other classical approaches emerged such as [77, 14, 15, 62], which some targeted the issue of OF in textured regions [14, 15], and others [4] fused it with intensity-images, these method were inconclusive and ineffective in computing the EV-OF due to two major limitations: 1) they can only compute the temporal derivative of the brightness, therefore more assumptions are needed to estimate the spatial derivative, 2) however due to the small amount of events fired as an edge crosses on it, it is hard to recover the spatial gradients which is assumed locally smooth.

Some EV-OF methods [43, 12] considered the intensity as an extra unknown in the gradient constraint equation, therefore proposed to jointly solve for the OF and intensity. They modified the global BCE formulation by adding smoothness on the intensity and a data term to account for the time input coming from AE data. The latter is solved as a minimization problem

over a Discretised local image structure as an event frame in a fixed time window. Nagata et.al [122] corrected the OF from [12] by using the 2D motion field equations to solve for the focus of expansion (FOE) to correct the OF estimation along the radial direction, which the smoothness loss does not account for. This approach requires the estimation of the yaw rate from event frames (angular velocity perpendicular to the image plane) prior for solving the FOE. Pan et al. [135] presented a more robust way to jointly estimate the OF with a single blurred imaged and its corresponding accumulated events. They improve the brightness Constancy equation to encode the real intensity, and added a blurness equation. Their approach handled rapid motion and blur batter then some existing EV-OF. These methods require handcrafted spatial and temporal regularizers (in the form of smoothness assumptions), which limited the quality of the OF estimation them sensitive to the smoothness assumption where they needed many events to fire to accurately estimate the OF.

4.1.2 Variational Approaches

Variational approaches that consider the local distribution of events in x-y-t space, are more robust and preferred over the classical approaches. This stream was initially introduced by Benosman et al. [18]. This category is more general then the BCE-based one, since it relies on the local distribution of events spatially in the image, in terms of time surfaces(see Eq. (4.3)). The original algorithm introduced what is known by surface of active events (SAE), which describes the movement of an edge firing events along it that fits points on spatio-temporal surface. The OF is estimated by fitting a plane within an accumulated time window on the structured image, where they prove that the slope of the surface in the plane encodes the edge motion. The accuracy of the OF depends on the accumulated time window of the fitted plane and how it evolves with time. Further, it only estimates the OF perpendicular to the edge. Later on Akolkar et.al [1] addressed the limitation

the aperture problem in this plane fitting approach, by using the fitted OF to correct for the true OF along the direction of the edge. Almatrafi et al. [3] addressed the lack of robustness to rapid motion from the plane fitting approach, by deriving an exact OF without Taylor series approximation, from a distance encoded event frames. Fei Low et al. [104] simplified the mathematical formulation of the plane fitting approach and added a greedy algorithm to optimize the accumulation of events. This aimed at speeding up the plane fitting approach at the expense of losing some OF estimation accuracy. These approaches showed that they are capable of outperforming the classical frame-based approaches. However, they all suffered from the goodness of the spatial window fit in the encoded image structure, which violates the spatial smoothness assumption if the window is too large or too small.

Mueggler et.al [116] went a step further by improving the plane fitting in the encoded image structure, predicting the time when future events will fire using τ in a regularization scheme. The results on EV-OF outperformed the original plane fitting method that relies on a constant event-accumulation interval. Stoffregen and Kleeman [157] used the lifetime estimation τ to create a different plane fitting technique: all events belonging to the same object have an estimated flow plane based on the structure of that object, therefore each event is segmented to create a new flow plane if it does not match to an already existing structure, otherwise, it is assigned its OF is computed based on the existing plane. The established planes are updated dynamically using the lifetime estimation τ. Gallego etl.al [60] estimate EV-OF by producing motion compensated event frames. They accumulated events into an event frame structure, then applying frame-based image warping techniques in order to maximize the sharpness of the event frame. They pass an adaptive filter on the event frame in the spatio-temporal direction which gives the best filter response. Along the same criteria, Zhu et.al [179] used motion compensation by accumulating an event frame over an optimized time window

using [116], to compute an average OF between the accumulated EF dictated by the lifetime of events. Reinbacher et al. c[144] tackled the the issue imposed by spatial window by doing spatial smoothness regularly at periodic times, which was was found to deteriorate the motion estimation accuracy resulting in blurred over smoothed or under smoothed reconstructed images when compared to spatially free approach [150]. Khoei et.al [85] created a 4 stage model to simulate a lateral geniculate nucleus (LGN) filtering to estimate OF. It consists first of denoising the output of an event camera, then they track the activity of event via a layer grid to model a simple LGN. the layers grids are encoded in a 140ms which is 4 times the actual frame-based rate. None of these approaches addressed the impact of the spatial smoothness window.

The variational stream show more accuracy and robustness over the classical approaches [148]. They also showed that they are capable of outperforming the classical frame-based approaches [18, 157], However, their OF estimation is highly affected by the spatial distribution of events within their accumulated frames, where too small or too large windows violate the smoothness assumption, which deteriorate the OF accuracy since events are meant to fire along object boundaries.

4.1.3 Deep Learning Techniques

Recently, deep learning approaches started to emerge. They rely on the availability of large amounts of event data to be accumulate them into event frames in order to apply frame-based ANN. The unsupervised learning approaches from frame-based cameras, namely Back-to-basics [81] and UnFlow [111], offered a new paradigm of OF learning to overcome the lack of labeling in datasets. Such methods were motivational for learning event-based motion estimation, where labeled datasets are rare.

These methods, were used as the base of the first EV-OF learning approach, called EV-Flow-Net [180]. Their main reasoning behind the lack of

success of deep learning for event cameras in general, was that these sensors do not output an image. Therefore, they based their idea on encoding events into an image structure to feed it directly to a CNN. They transformed events into a 4-channels encode event frame accumulated over a window of time. The first two channels count the number of positive and negative events respectively, while the last two channels encodes the timestamps by storing the last positive and negative timestamps respectively. The input event frame is fed into a CNN to estimate the optical flow as seen in Fig 4.1. They used a type of event cameras called DAVIS [22]. DAVIS outputs grayscale image synchronized with event data, at the standard regular camera frame rate. EvFlow-Net used a pair of grayscale images occurring before and after the encoded frame, i.e. the encoded frame consists of accumulating the events that occurred between those two frames. Its input also consists of the pair of grayscale images, to compute a self-supervised loss. The architecture is a similar encoder-decoder from [81] while relying on the classical photometric loss. When compared to the frame-based approach UnFlow [111], the results seemed to be in favor of UnFlow. Their encoded event frame method, besides only operating at low frame rate, fails in the presence of dense and large movements. Later on, they improved EV-Flow-Net [181] by using a contrast maximization loss measuring the consistency of the OF estimation across voxel-grids, where their results were on par with [111]. An important observation in [181] was that the spatial smoothness constraint used to combat data sparsity needed for creating event frames, tends to blur object boundaries due to the deterioration of accuracy in estimating the motion at object boundaries.

Evenly-Cascaded neural Network (ECN) [178] is another existing approach for learning EV-OF. It replaces the last 2 channels in the EvFlow-Net event frame generation, by one channel that take the average timestamp of all the events fired on that pixel during the fixed time window. The main aspect of their approach is jointly estimating optical and camera's egomo-

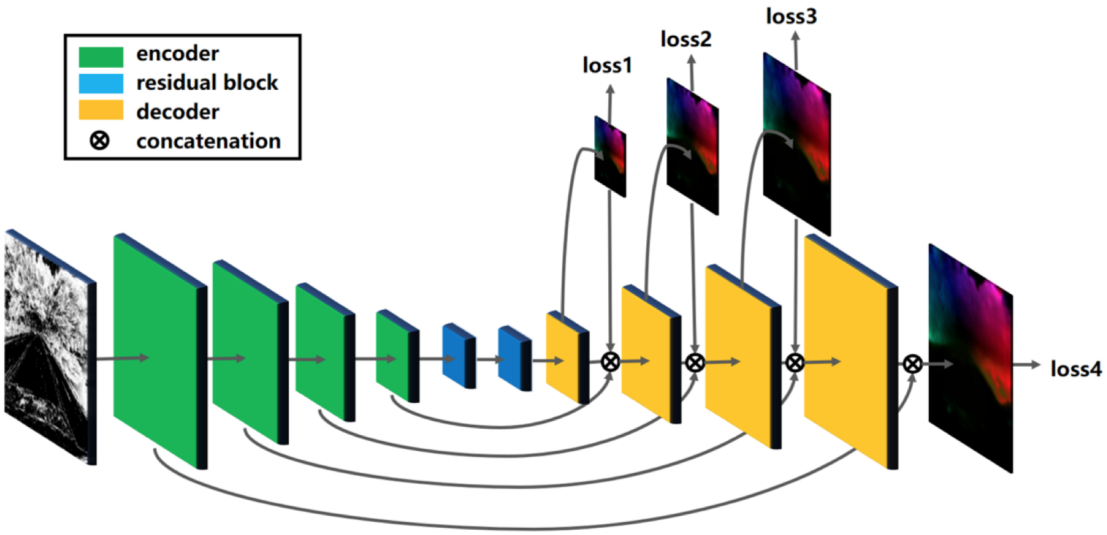


Figure 4.1: EvFlow-Net architecture. Figure taken from [180].

tion. Their ECN, illustrated in Fig. 4.2, consists of 2 networks: a camera velocity network, and a depth network. The input to the ECN is a set of three consecutive encoded frames. The camera pose network consists of an encoder that will predict the relative velocity pose vectors with respect to the middle event image. The depth network uses only the middle event frame as its sole input, to predict the depth. Given the predicted depth and the camera velocity vectors for neighboring frames, the OF is estimated. ECN uses the spatial smoothness loss. Therefore, ECN is also an unsupervised EV-OF network.

Kepple et al [84] proposed a CNN that jointly learn the camera pan and tilt rates and the OF. They proposed 2 CNNs, a visual motion network which predicts the camera motion from 4 event frames accumulated over milliseconds window. To overcome the aperture problem posed by this estimation they proposed a joint second CNN that measures the confidence of the predictions. They achieved faster computational speeds then EV-Flow-Net but

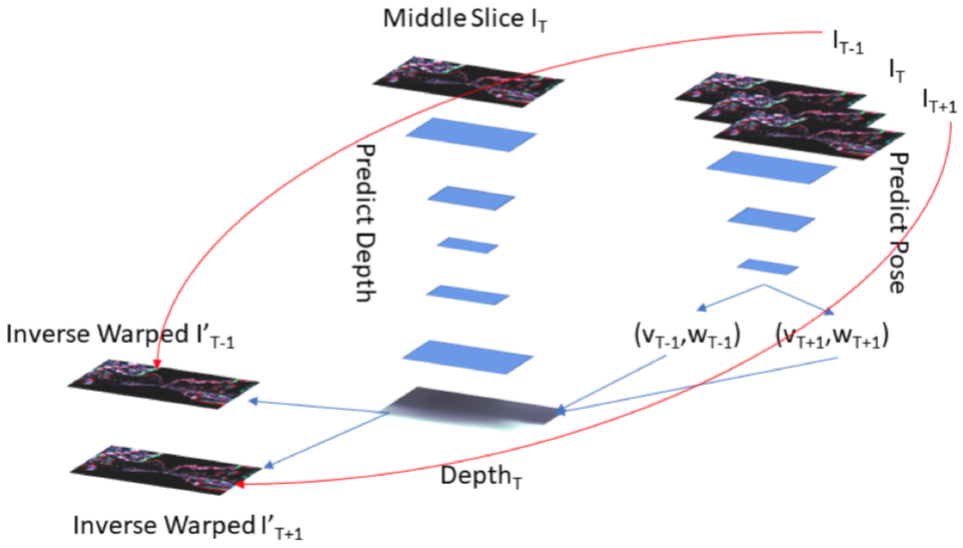


Figure 4.2: ECN architecture. Figure taken from [178].

less accuracy. Paredes-Valles and de Croon [136] proposed to encode events into voxel-grids structure, which are then fed to two separate networks. The first is FlowNet [51] and the second one is ReconNet. The approach is self supervised, Their main contribution was integrating the contrast maximisation loss from [181] to train and compare the resultant event frame from both networks with the warped event frame from the OF estimated.

Gehrig et al. [65] presented another synchronous approach which was a combination of three main events representation. Their major finding was in comparing their results against other ways of accumulated grid-like representations, and showed that even within the main representations used to process events in accumulated grid-based methods the accuracy of the OF estimation was very sensitive to the type of grouping used.

In the frame-based field, CNN-based OF approaches FlowNet3.0 [79] and PWC-Net [161] have largely outperformed hand-crafted OF approaches.

Such a success is yet to materialize in motion estimation from event cameras. Furthermore, these approaches are not clearing showing tangible benefits from using address event data over regular images. The reason behind that is due to the use of Event Frames to extend frame-based objective functions to work on the generated event frames, which effectively eliminate most of the potential advantages of event cameras.

4.2 Event-based Egomotion Estimation

Egomotion is the process of estimating the motion of a camera in 3D scenes. Solving the egomotion problem with event cameras in its most general setting i.e. 6-DOF in natural 3D scenes, is a challenging problem due to the data association and ill-posedness challenges. For this reason, event-based egomotion approaches addressed it step by step with increasing complexity such as the type of motion (2D pure rotational motion or planar motion, 3D general motion in natural scenes), depth consideration, or additional requirements (external sensors, requirements on scene types). Table 4.1 classifies the main related work using these complexity groups, and highlights how our approach will be fundamentally different. We now review the relevant work.

The early egomotion approach from event cameras was presented by Weikersdorfer et al. [171, 119]. They designed a likelihood function for a particle filter that quantifies the observed events given a map of the scene and some information about the camera pose. It only works on planar scenes parallel to motion plane generated from artificial line patterns. They extended their approach in [170] to 3D camera pose estimation in general scenes, however they required a depth map from RGB-D sensors [127], forcing the approach to become synchronized with fixed frame rate, by encoding events into an event frame to track the evolution of its local structure.

Later approaches increased the complexity of the problem, by either using generic handcrafted approaches, or relying on additional sensors. Cook et

Table 4.1: Event-based methods for egomotion estimation. The type of motion is labeled "2D" (3-DOF motions, e.g., planar or rational) and "3D" (6-DOF motion in 3D scenes). Columns indicate whether the method considers depth in the scene ("Depth"), are free from Local Image Structure ("LIS free"), are free from Spatial Smoothness (SS) assumption ("SS free"), and any additional requirements. Note that the work of this thesis will be fundamentally different.

References	**Dim**	**Depth**	**LIS free**	**SS free**	**Additional requirements**
Cook [43]	2D	✗	✗	✗	rotational motion only
Censi [34]	3D	✗	✗	✗	attached depth sensor
Kim [87]	2D	✗	✗	✗	rotational motion only
Kueng [90]	3D	✓	✗	✗	intensity images
Rebecq [143]	3D	✓	✗	✗	-
Reinbacher [144]	3D	✓	✗	✗	intensity images
Gallego [62]	2D	✓	✗	✗	rotational motion only
Liu [99]	2D	✓	✗	✗	rotational motion only
Reverter [147]	3D	✓	✗	✗	-
Peng [138]	3D	✓	✗	✗	-
Wang [168]	3D	✓	✗	✗	-
This work	3D	✓	✓	✓	-

al. [43] presented a handcrafted synchronous approach that jointly estimate the egomotion, optical flow, and intensity. This method is very generic enforcing a rotational constraint motion only. The handcrafted work in [34, 90] estimated the camera translational displacement synchronously for planar scenes. Their approach required the fusion of a frame-based camera to the event cameras to estimate small translational displacements between event frame and regular frame. Handcrafted spatial smoothness was used, while this system was restricted to operate at the frame rate of the frame-based cameras. Their results were slightly worse then frame-based approaches. Muggler et al. [119, 118] tracked the camera pose for very rapid motion by

a hand-crafted approach requiring synthetic 3D black and white maps. At first, they brute forced the method by tracking events firing synthetically one by one next to each other on the map, by using PnP frame-based methods. They further extended their approach to estimated the camera trajectory instead of the camera pose, using a reprojection frame-based loss. Gallego et al. [61] used a probabilistic Bayesian filter to track the 6-DOF of the camera. The filter requires an existing 3D map of the scene to compute a likelihood function based on mixture of densities. Bryner et al. [28] also presented a synchronous approach given a 3D map of the scene, by applying non-linear optimized frame-based reprojection loss between an event frame, and intensity and depth images from traditional camera derived from the given 3D map. Similarly, Reinbacher et al. [144] used a mapping and tracking algorithm to estimated the camera 3D rotation via a frame-based photometric loss restricted to rotational motion. Chamorro et al. [37] used a Lie-EKF filter to perform fast event to line matching using event frames, grayscale images, and prebuilt 3D map to track the camera pose. These methods have inconclusive results compared to image-based approaches, although they outperformed them specially in challenging scenes. They do not exploit the event cameras' advantages at all since they require additional external information, and they were sensitive to the smoothness constraints assumed to track the evolution of their local event frame structure.

Approaches that narrowed down the motion estimation to 3D pure rotation gained success by showing materialized results against the frame-based approaches. Kim et al [86] used a particle filter to estimate the 3D camera rotation state for high dynamic range 3D image reconstruction. They assume spatial smoothness to estimate a brightness gradient map of the scene from the 3D rotation of the camera. The filter requires a 3D (panoramic) map of the scene to estimate the camera rotation, only works for rotational motion, and the accuracy of the filter to estimate the rotational state is not reported. They extended this approach in [87] to jointly estimate the camera pose,

depth of the scene, and the intensity image. They replaced their particle filter with an EKF that is joint with 2 other probabilistic filters. Similarly to [86], their filtering requires intensity images and GPU to reconstruct depth maps, which make it only suitable for offline applications. Rebecq et al. [143] further extended the work in [86, 87] by replacing the depth estimation in [87] with a 3D simultaneous mapping through 3D reconstruction and map alignment assuming smoothness constraint around the encoded image structure. Their results were on par with regular images. However, these approaches have limited robustness due to the simple assumptions made on the camera pose to track the evolution of the local structure for data association. They also require additional information which are typically provided by frame-based sensors, 3D map of the scene, or intensity images to solve the data association.

Recently, contrast Maximisation (CM) approaches have been seeing major success in event-based egomotion. The idea behind using a contrast maximization technique was to associate events that produce sharp edges in the encoded local structure when warped using locally smooth velocity, solving for the data association by minimizing the evolution of the local image structure. 3D rotational approaches are currently providing direct comparison against each other for egomotion estimation using event cameras. Gallego et al. [63, 62] estimated the camera angular velocity [63] and heading [62] by warping each event based using a constant angular velocity model, then encoding the warped events into an event frame. Their original approach in [63] was a synchronous 3D rotational speed that relies on maximizing a frame-based contrast objective function applied onto their generated event frames to track the evolution of the local structure. They extended their approach in [62] to estimate depth using a frame-based multi view stereo planar homography approach, before applying their maximization of contrast to estimate the depth. Xu et al. [176] used the method in [63] to apply frame-based image maximization energy with smoothness constraint

on event frames, in order to track the camera's pose. Similarly, Nunes et al [132] improved the method in [62] by replacing the maximization contrast by a frame-based entropy minimization framework, for 3D rotation and camera heading estimation in planar scenes. Another improved extensions were presented by Liu et al [99] where they proposed to add a branch-and-bound (BnB) to perform a global optimization on the contrast maximization approach in [99] to estimate the camera's angular velocities. They latter added a simple spatiotemporal registration (STR) approach in [100] assuming rotational motion only. This approach is currently one of the best 3D rotational estimation approaches outperforming [62, 63, 176].

Finally, the general motion estimation from event cameras was considered. The contrast maximization on event frames was expanded to deal with general motion estimation via Branch and Bound [138, 168]. Another major stream of general motion estimation from event frames, relied on camera's pose estimation techniques such as standalone traditional keyframes methods [147], or aided by external sensors (IMU or grasycale cameras) [117, 166, 91]. These approaches seem to be working better then frame-based techniques, in situation where there was rapid motion, motion blur, and high illumination changes. However, CM and pose estimation techniques for event-based egomotion are sensitive to the event frames structure, where the tracked evolution of those structure violates the local smoothness assumption in many situations, and some assume pure rotation in scenes where significant translation motion is present.

4.3 Event-based Objective Functions

We saw in the previous sections how motion estimation approaches with event cameras are performing, and identified key gaps that could help develop more improved motion estimation approaches. This section goes into some of specifics of the objective functions of the main approaches to pro-

vide a better understanding of those approaches and their limitations on event-based motion estimation. They can be classified into the following two categories:

- Classical objective functions that rely on frame-based formulation.
- Time Surface (TS) or variational-based objective functions.

4.3.1 Classical Objective Functions

In event-based vision there are no intensity information, therefore a formulation of intensity from event cameras was the major focus of these approaches. Therefore, the classical stream adapts objective functions from frame-based cameras. Benosman et al. [19] introduced a BCE-based approach. This method adapts the Lucas and Kanade local smoothness assumption. Their original derivation which is asynchronous by definition, inspired the later work in this category. It consists of defining a neighboring window around a local structure in the image, to compute the spatial and temporal derivatives, therefore the gradient constraint equation is written as:

$$\sum_{t-\Delta t}^{t}(e(x_i,y_i,t)-e(x_{i-1},y_i,t))u=\frac{1}{\Delta t}\sum_{t-\Delta t}^{t}e(x_i,y_i,t), \tag{4.1}$$

$$\sum_{t-\Delta t}^{t}(e(x_i,y_i,t)-e(x_i,y_{i-1},t))v=\frac{1}{\Delta t}\sum_{t-\Delta t}^{t}e(x_i,y_i,t), \tag{4.2}$$

Brosch et al. [23] added a second order term to the temporal gradient estimate in Eq. (4.1) and (4.2), to make the derivative estimation more stable. Bardow et al. [12] considered the intensity as an extra unknown in the gradient constraint equation, therefore proposed to jointly solve for the OF and intensity. They added spatial smoothness on the intensity and a data term to account for the time input coming from AE data over accumulating

them over a voxel grid. The latter is solved as a minimization problem over a Discretised event frame in a fixed time window. Barranco et al. [14, 15] argued that in the case of textured edges (or contours) the smoothness assumption will fail. Therefore, they offered an alternative by slicing the event frame accumulated over a time window, then used cross-correlation between the slices assuming spatial smoothness. Along the same line, Cook et al. [43] presented a simpler non-traditional algorithm to jointly estimate the intensity and OF, within in an interacting model, for only rotational movement. Weikersdorfer et al. [171, 119] proposed simple asynchronous objective functions that computes the reprojection error (assuming spatial smoothness) between the event's location and the closest edge in a map of a scene. It only works for 2D translational scene with constant depth given the 3D map of the scene. Nagata et al. [122] introduced FOE-based regularization loss that works by intersecting of the translational velocity vector and the image plane. They used this loss to correct the spatial smoothness regularization loss in estimating the radial component of the optical flow. It requires the estimated yaw angular motion computed by accumulating events to reconstruct event frames using the loss from [12], applying frame-based loss to estimate the angular motion, then using it as an input to their FOE to regularize the OF estimation. It is inconclusive to say if these approaches outperform frame-based techniques given the easier instances of motion considered.

On the other hand, more successful frame-based objective functions started to find more success in event-based motion estimation. Contrast Maximization(CM) techniques [62, 99, 100, 138, 168] proposed to warp accumulated events into an image structure, which produces motion patches representing the brightness increment of the evolution of that structure(proportional to the gradient constraint) while imposing local smoothness, that corrects the motion estimation. Almatrafi et al. [4] used the BCE formulation to interpolation objective function assuming local smoothness, and find the gradient of grayscale images intensity, which then they use to correct the intensity com-

puted from intensity images with the intensity recovered from events frames accumulated at a fixed frame rate. Pan et al. [135] used the BCE equation for events in log space to encode the motion and events relationship, conserving the original absolute intensity. They also added a blur objective function to the smoothness terms. These approaches showed successful motion estimation even in very fast motion scenarios. However, their CM formulation is sensitive to the evolution of the local structure of the event frame, where low rate of events firing (sparse scenes only on object boundaries) violate their local smoothness assumptions which dictates their CM formulation.

Gallego et al. [60] aimed at studying the alignment of events in the local image structure for motion compensation. Towards that they tested about 20 frame-based objective functions (used in unsupervised learning) on an motion-compensated event frame. The event frames are warped using image warping techniques, and then the 20 objective functions were applied over the optimized event frames. Their results showed that most of the frame-based objective functions can be applied to AES applications such as OF and egomotion. However, they highlighted that the stream of objective functions that focus on adapting frame-based objective functions eliminate most of the advantages the event cameras offer.

Recently, deep learning techniques adopted classical unsupervised losses from the successful learning approaches in regular cameras. The main approaches [180, 178, 136] used the photometric loss and smoothness loss, exactly as defined in [81], which has been enjoying a great success in frame-based learning approaches. The initial approach [180] had an objective function which computes the error between grayscale images occurring before and after the event frame structure, i.e. the event frame consists of accumulating the events that occurred between those two frames. They apply their loss by backwarping (photometric loss based on the BCE) the event frames to track their evolution over time, then applying the smoothness loss. Stoffregen and Kleeman [158] used the contrast maximisation formulation from [181] to con-

duct an analysis using many frame-based objective functions in order to try and define a proper number of events to be accumulated within the voxel-grid. Their results showed the processing of events in accumulating them into grid-based groups play the biggest role in the performance of objective objective functions used. These approaches fail to generalize to different environments not seen at training time. The reason behind that is due to the use of event frames to extend frame-based objective functions based on the evolution of local those image structures over time (relying on the smoothness loss) to work on the generated event frames, which effectively eliminate most of the potential advantages of event cameras.

4.3.2 Variational-based objective functions

TS-based or variational-objective functions approaches that consider the local distribution of events in the local image structure in the x-y-t space have been showing more robustness then the classical approaches when it comes to engineered techniques for optical flow estimation in particular [148]. The variational objective functions stream rely on spatial and temporal smoothness assumptions, while requiring accumulating events into some sort of an image strcuture in order to capture the local spatio-temporal relationship between events.

The original formulation was introduced by Benosman et al. [18] and is now know as the surface of active events (SAE) $\Sigma_e(x, y)$ defined as $\Sigma_e(x, y) = t$, where (x, y, t, p) are event data. $\Sigma_e(x, y)$ is a function which maps to each event position (x, y), a time t. Differentiating $\Sigma_e(x, y)$ results in predicting the mapping at $p + dp$ as function of the gradient. The velocities on the image plane are related to the gradient of the SAE by:

$$\nabla\Sigma_e(x, y) = (u(x, y)^{-1}, v(x, y)^{-1})^T. \tag{4.3}$$

Eq. (4.3) assumes a local velocity constancy in the form of spatial smooth-

ness. For each event, it fits a local plane by accumulating the neighboring events in a time window and study the evolution of this local structure over time. It uses LSM [64] to estimate a plane and loop over all events to make sure they take the ones only belonging to the plan. Finally, the plane's normal vector is used to compute the velocity of each event. This objective function have became of most commonly used methods in event cameras specially for motion estimation. The goodness of this method depends on the accumulated time window of the fitted plane to ensure that the velocity constancy is not violated: if the window is too large overfitting occurs, if the window is too small accuracy is lost. They typically accumulate events within 10ms (thousand times the event cameras latency). The accumulation of events idea to study the evolution of the local structure around these events started to evolve to become one of the major framework to event-based vision, and its major bottleneck at the same time. The original algorithm did outperform frame-based approaches for motion estimation.

Till this day, many event-based motion estimation approaches rely on this SAE objective function. Clady et al. [40] used the local plane fitting function to compute the time of contact between a robot holding an event camera and an object. The method expanded the local plane fitting by adding a probability map of the visual field. Akolkar et al. [1] addressed the limitation the aperture problem in this plane fitting approach, by adding an additional objective function that minimizes the normal velocity component estimated by the plane fitting function with respect to orientation of the edge and the true motion direction. They further extended the original plane fitting loss to work with complex object shapes [2] , by using the magnitude of the normal velocity component of the estimated plane, which is directly related to the orientation of the object's edge, and linearize it using multiple small edges to complete the object shape. Almatrafi et al. [3] presented a distance surface loss based on TS where they accumulate event into distance frames: each event frame encodes the spatial distance as the intensity value. Then

they use this distance frame to derive the OF as spatio-temporal gradient of the distance function, exactly like the plane fitting loss [40]. They claim their derivation is exact in contrast to the plane fitting one which rely on Taylor series approximation, which makes it more robust to fast motion. Eventually they use the classical frame-based photometric objective function. Their results were not better then the original approach. Fei Low et al. [104] improved the SAE by simplifying its mathematical formulation with the addition of a collinearity constraint. These approaches continued to have the same success as their parent objective function approach against frame-based approaches specially in situations where motion was fast, illumination changes were large, and motion blurness. However, they inherent the goodness of the accumulated window limitation due to the spatial smoothness assumption of this stream of functions.

Another main variant of the SAE was introduced by Mueggler et al. [116]. They went a step further by estimating the maximum time it would take for the brightness gradient at the current event location to trigger a new event in a neighboring pixel, in the vicinity of 1 pixel radius (see Fig. 4.3). The latter is called lifetime, τ, of an event and is defined by Eq. (4.4):

$$\tau(x, y) = \|\nabla \Sigma_e(x, y)\| = \sqrt{\frac{1}{u^2} + \frac{1}{v^2}}. \tag{4.4}$$

They can optimize the accumulation time window of events based on $S\tau$, where S is the desired displacement within this time window. Their results outperformed the original TS approach [18] and image approaches.

Stoffregen and Kleeman [157] used the lifetime estimation in Eq. (4.3) to create a different plane fitting algorithm, where all events belonging to the same object have an estimated flow plane based on the structure of that object, therefore each event is segmented to create a new flow plane if it does not match to an already existing structure, otherwise, it is assigned its OF is computed based on the existing plane. The established planes are updated dynamically using the lifetime estimation τ. Zhu et al. [179] used

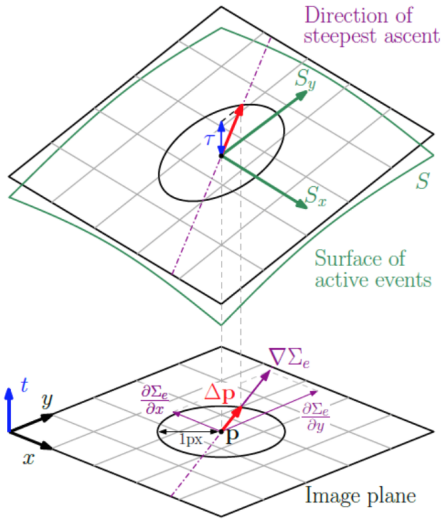


Figure 4.3: Life time of event visualization as the planar approximation of the SAE. Figure taken from [18].

Eq. (4.4) to accumulate event frames based on the lifetime depicted by the length of the OF. They created a spatio-temporal OF sparsity constraint loss assuming that a constant average velocity withing the accumulated time window. The loss minimizes the data association between the events within the event frames if the average OF withing the accumulated time window is correct (similar to backward image warping). These methods are sensitive to the estimated lifetime of events which dictates the milliseconds window over which the constant average velocity is not violated. Although they improved the goodness of the fit of the accumulation window, it continues to be violated at object boundaries due to the spatial smoothness [150].

Zhu et al [181] replaced their photometric loss in [180] by a TS loss based on a voxel grid image structure which is minimized by summing across event polarities that projects that local structure back into a single average time window where the average OF is consistent. This function, inspired by the contrast maximization objective function, is very similar to the plane fitting

loss [18], in assuming that the flow is inversely proportional to the gradient of a grid-representation of the events within a spatio-temporal accumulated time window. Kepple et al [84] proposed a TS objective function which takes 4 event frames voxel-grids to minimize the camera tilt and pan rates with respect to the OF. This supervised loss requires prior information to aid that joint estimation. They handcrafted it to fit their 2 proposed CNNs with precise ground truth for the camera motion to recover the OF. These approaches fail to generalize to different environments not seen at training time. Although they are showing better motion estimation compared to frame-based techniques, they are susceptible to the accumulation of events as noted in the study of [65]. Furthermore, the work of [181] noted that their approach tends to blur object boundaries due to the spatial smoothness constraint, which can significantly deteriorate the motion estimation accuracy at object boundaries.

4.4 Filtering In Event Cameras

The previous sections show that most of the event-based motion estimation approaches tend to aggregate events into frames, and as shown in the analysis work [65] it is not trivial what is the best way to bundle events into frames, and concluded that to exploit event cameras to their full potential, asynchronous processing is the way to go. Furthermore, one of the main premises of this thesis is to solve the motions estimation from event cameras *asynchronously*. In this section, we review the related work on filtering techniques in event cameras as they have the capability of processing event-by-event.

In general, asynchronous processing with event cameras is commonly done through either the use of filters or the use of neuromorphic methods. It is worth noting, that the latter rely on a new class of neuromorphic hardware processors such as the IBM TrueNorth [112], which are asynchronous by

design. A number of successful work was based on the use of Spiking Neural Networks (SNN) [134, 133, 101, 71], which are biologically inspired networks, and take the temporal information from spikes making it useful for the event cameras. However, these hardware processors are not yet well established, which means that the processors must overcome the same entrenched competitor practicality problem that the event cameras are facing with the frame-based cameras. Additionally, SNN's major drawback of having their objective function not differentiable, makes back-propagation inapplicable. Therefore, until neuromorphic methods become a mainstream, event cameras algorithms continue to make use of the advantages offered by parallel computing similarly to frame-based camera. Thus, neuromorphic methods are not studied in this thesis.

There are two main types of filtering techniques used in event cameras:

- Deterministic Filters.
- Probabilistic Filters.

4.4.1 Deterministic Filters

Brosch et al. [23] applies a set of state-of-the-art deterministic-selective filters (equivalent to spatio-temporal correlations filters in frame-based approaches) or spatio-temporal filters on accumulated event streams. The filters are hand crafted to select different motion speeds and directions. Smoothing temporal deterministic filters such as [150, 145] presented a temporal filter to smooth events for image reconstructions and fusing events into frames for 3D image reconstruction. The filter can update each pixel intensity every time an event fire lead to reduced noise on the reconstructed event frames. Scheerlinck et al. [152] presented a brightness deterministic filter for spatial image convolution from events, for better 3D image reconstruction. Such methods have weak or no motion assumptions therefore do not adapt well to motion

estimation. They rely on spatial smoothness assumption, which is sensitive to differentiation leading to inconclusive results.

4.4.2 Probabilistic Filters

Probabilistic Filters [163] are the dominant stream for asynchronous processing with event cameras. These filters naturally fit the motion estimation, since they rely on motion assumptions, therefore can be used to design likelihood functions that adapt to AE data. All the previous work that use probabilistic filters, *focus on designing likelihood function* based on the event generation process.

Censi and Scaramuzza [34] used a Bayesian filter with a simple likelihood function which fuses an event frame with a regular intensity frame (a requirement for this filter). Gallego et al. [61] use a probabilistic Bayesian filter to track the 6-DOF of the camera. The filter requires an existing 3D map of the scene to compute a likelihood function based on mixture of densities. Chamorro et al. [37] present a Lie-EKF filter to handle the derivatives and covariances better then a standard EKF. However, it processes event via event frames, and requires the use of grayscale images for initialization given a predefined 3D map.

Simple Gaussian filters approaches [96, 46, 42, 53, 140, 131, 44] were initially used to detect blobs of events, where they associated fired events with their nearest blob for tracking. Weikersdorfer et.al [171, 119] used the first particle filter in event cameras. Their filter was restricted to either only work on 2D planar scenes with constant depth, or required a depth map through external RGB-D sensors [127]. Similarly, approaches that track simple generic shapes such as circular shapes. [67, 69, 165] used simple Bayesian filters to track circular shapes in scenes by accumulating events into frames. Kim et al [86] used a particle filter to estimate the 3D camera rotation state for high dynamic range 3D image reconstruction. Its likelihood function relies on the BCE assumption to estimate a brightness gradient map of the scene

from the 3D rotation of the camera. The filter requires a 3D (panoramic) map of the scene to estimate the camera rotation, only works for rotational motion, and the accuracy of the filter to estimate the rotational state is not reported. They extended this approach in [87] to jointly estimate the camera pose, depth of the scene, and the intensity image. They replaced their particle filter with an EKF that is joint with 2 other probabilistic filters. Similarly to [86], their filtering requires an intensity image. Zhu et al. [182] used an EKF fused with an IMU to estimated the correct for the camera motion by designing a new likelihood projection function. However, these approaches have limited robustness due to the assumptions made on the camera pose, and require additional information.

Recently, asynchronous particle filtering approaches [5, 6] for feature tracking using multi-hypotheses state estimation. This latter stream of work show the importance of asynchronous processing in exploiting event cameras to their full advantages, when compared to accumulating events into grid-like representations.

4.5 Conclusion

All event-based motion estimation approaches rely on the fact that events encode the local structure of the imaged scene. Then, they estimate motion by tracking the evolution of the image structure over time, which is commonly achieved through developing motion-based objective functions addressing the data association challenge. The major problem in this way of estimation is that events are only an approximation of the local structure that can be very sparse in some cases. Furthermore, spatial smoothness is commonly assumed to solve for the ill-posedness nature of the problem, which is easily violated along object boundaries or edges specially in the dominant stream that focus on aggregating events to synchronously track its evolution of the image structure. In this thesis, we want to avoid these assumptions, therefore, we

approach the problem in fundamentally different way, by developing new voting based-method that handle each event individually without assuming any relationship to its neighbors beyond the common camera motion. In the following chapters, we validate that such methods are sufficient for correct data association of events and their previous firings along the corresponding event tracks, resulting in more accurate and robust event-based motion estimation.

Chapter 5

Asynchronous Planar Motion Estimation

5.1 Introduction

In this chapter, we show that consistency with a single camera motion, where events constitute a single event track, can be used to solve the data association problem Eq. (3.6), then provide a framework for motion estimation based on that. We validate this hypothesis mathematically, then experimentally on a simplified motion case as a first step in solving the general form of the motion estimation problem Eq. (3.4).

Towards that, we create a new voting based method to handle each event individually. It relies on an exhaustive search over all possible event candidates to select those consistent with a single camera motion. We first provide a mathematical analysis to prove the sanity of this method. Furthermore, we exploit the voting method as the likelihood function in a particle filter framework for a simple planar motion case where a camera is translating parallel to a wall.

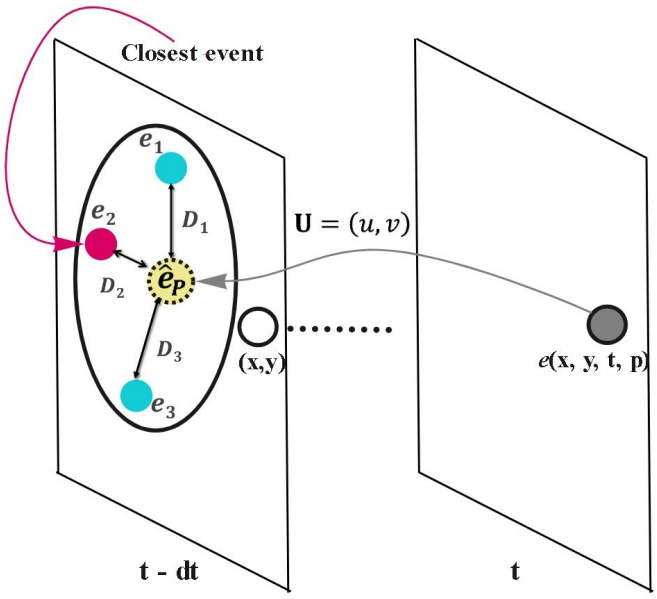


Figure 5.1: Graphical outline of the method. For an input event e, given a certain velocity $\mathbf{U}$, predict its previous firing $\hat{e}_p$ by backward projection. A spatiotemporal search is performed over all possible events candidates around $\hat{e}_p$ to select the ones that are consistent with the camera motion.

5.2 Method

Assuming the BCE and the constant velocity model, and in the presence of spatial discretization, noise, and uncertainty in the estimation of $\mathbf{U}$, the backward projection of an incoming event e can be defined as:

$$\hat{e}_p = \Pi(\mathbf{U}, e) \tag{5.1}$$

This event is predicted to occur at time $t - m\,\delta t$ and location $(\hat{x}, \hat{y})$. Note that predicted events occur over a smooth (continuous) spatiotemporal domain, while measured events occur over a discretized spatiotemporal domain. We will dub predicted events with hats.

Let $\mathcal{E}_p(\hat{e}_p, n, m) \in \mathcal{E}$ (see Fig. 5.1) represent the set of measured events

within an $n \times n$ spatial window centered around $\hat{e}_p$, and a temporal window $m\,\delta t$. Further, we define a generalized distance metric D encompassing the spatial, time, and polarity 'distance' between events e_i and e_j as follows:

$$D(e_i, e_j) = \begin{cases} \min(r_t|t_i - t_j| + ||\mathbf{x_i}, \mathbf{x_j}||, d_{max}); & p_i = p_j \\ d_{max}; & \text{otherwise} \end{cases} \tag{5.2}$$

where r_t is a time penalty that addresses the ambiguity caused by uncorrelated events spatially close to the predicted event $\hat{e}_p$ but temporally far away from it and d_{max} is the maximum spatiotemporal distance parameter. It is also assigned to penalize events with different polarities ($p_i \neq p_j$) or those that project outside the image boundaries.

A search of all candidate events within the spatiotemporal window is carried out to select those consistent with camera motion. To define an objective function, we use a voting method based on the minimum generalized distance between $\hat{e}_p$ and the candidate elements of $\mathcal{E}_p(\hat{e}_p, n, m)$ as:

$$L(e, \mathbf{U}) = \min_{e_i \in \mathcal{E}_p(\hat{e}_p, n, m)} \Big(D(e_i, \hat{e}_p)) \Big), \tag{5.3}$$

Hence, Eq. (5.3) minimizes the BCE over all the previous events in a neighbourhood to solve the data association problem Eq. (3.6).

5.3 Framework

The aim is to exploit the voting objective function to estimate motion asynchronously from Eq. (3.4). We employ a particle filter as a straightforward sequential Bayesian way to estimate a simple planar motion. A particle filter suits our distance-based objective function since it involves a discrete search. The basic idea is, as soon as a new event is fired, to use our method as the likelihood that votes the velocity which is most consistent with the camera motion along the associated event track. The framework is described schematically in Figure 5.2 and procedurally in Algorithm 1.

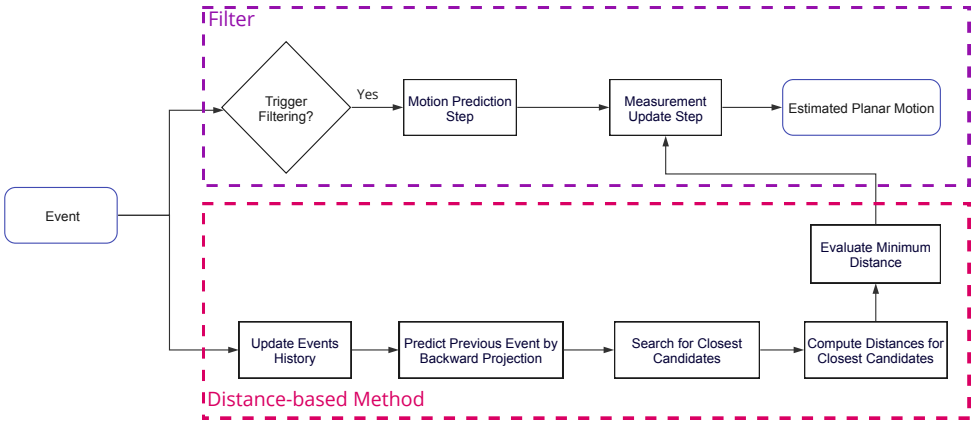


Figure 5.2: Schematic outline of the framework. It runs on an event-by-event basis as soon as an event is fired. Our distance-based method is integrated as the likelihood in a particle filter framework to vote the velocity which is most consistent with the camera's planar motion.

For planar motion scenes where $T_z = 0$ and $\mathbf{\Omega} = 0$, the equations of motion (3.4) reduce to:

$$u = f\frac{T_x}{Z} \tag{5.4}$$

$$v = f\frac{T_y}{Z} \tag{5.5}$$

For an event e, the filter is represented by a set of particles, $P^{(t)} = \{P_1^{(t)}, P_2^{(t)}, ...P_N^{(t)}\}$. Each particle consists of the current state $\mathbf{U}_i^{(t)}$ and its weight $w_i^{(t)}$, where $1 \leq i \leq N$. Note that estimating the OF $\mathbf{U}_i^{(t)}$ is equivalent to estimating the camera's 2D translation velocity $\mathbf{T}_i^{(t)}$.

We use the constant motion model to propagate the states of the particles:

$$U_i^{(t)} = U_i^{(t-\tau)} + \mathcal{M}(U_i^{(t)}), \tag{5.6}$$

where $U_i^{(t-\tau)}$ is the previous state of particle i at $t - \tau$, $\mathcal{M}(U_i^{(t)})$ is sampled from a Gaussian distribution in the x and y directions independently such that $\mathcal{M}(U_i^{(t)}) \sim N(0, \sigma_i^2)$, and σ is the standard deviation.

In the measurement update step, we use the measured event e to calculate the likelihood $P(e \mid U_i^{(t)})$ for each particle in the form of an exponential decay function:

$$P(e \mid \mathbf{U}_i^{(t)}) = \exp(-\alpha_p L_i^{(t)}(e, \mathbf{U}_i^{(t)})), \tag{5.7}$$

where α_p is a decay scaling parameter and $L_i^{(t)}$ is our distance-based function Eq.5.3. We update the weight w_i of every perturbed particle by applying the standard Bayes rule:

$$w_i^{(t)} = P(e \mid \mathbf{U}_i^{(t)}) w_i^{(t-\tau)}, \tag{5.8}$$

where $w_i^{(t-\tau)}$ is the weight at the previous timestep $t - \tau$. The weights are normalized according to

$$w_i^{(t)} = \frac{w_i^{(t)}}{\sum_i^N w_i^{(t)}}$$

Systematic resampling [52] is carried out when the effective number of particles

$$N_{eff} = \frac{1}{\sum_{i=1}^N (w_i^{(t)})^2}$$

drops below a threshold such that $N_{eff} \leq \frac{N}{2}$. The OF $\mathbf{U}$ is estimated as the weighted average over all the particles.

Algorithm 1: Asynchronous Planar Motion Estimation

Input: Current Event

Output: Optical Flow

Initialize $N \times 2$ particles (velocities) with uniform weights;
if $p > 0$ **then**
    Add and update the positive polarity history;
else
    Add and update the negative polarity history;
end
if *enough events* **then**
    for *each particle in* N **do**
        Motion prediction step using Eq. 5.6;
        Predict the previous event $\hat{e}_p$ by backward projection Eq. 5.1;
        Search for the closest events around $\hat{e}_p$;
        Compute the distance cost for the closest events using Eq. 5.2;
        Evaluate the cost using Eq. 5.3;
        Use the returned cost to compute the likelihood using Eq. 5.7;
        Update the particle's weight using Eq. 5.8;
    end
    Normalize the weights of all particles;
    Resample if $N_{eff} \leq \frac{N}{2}$;
    Output the mean OF of all particles;
end

5.4 Mathematical Validation

In this section, we validate mathematically that minimizing the BCE over all previous events in a neighbourhood can be used for data association and motion estimation.

5.4.1 Notations

Given a point x_i moving at a constant velocity v along a line on the camera projection plane. Its position at time t is given by:

$$x_i(t) = v\,t + d_i$$

where d_i is the position of x_i at time 0. We assume that:

- The camera plane is infinite and discretized into pixels.
- The point represents the center of an object covering an area equivalent to the size of a pixel.
- An event is generated at a pixel when x_i reaches the edge of that pixel, such that half of the pixel would be covered by the point.

With the above assumptions, events are generated at time t for all points satisfying:

$$C(i,t) = v\,t + d_i - floor(v\,t + d_i) = 0 \tag{5.9}$$

It follows that events will be generated with a periodicity of $p = \frac{1}{v}$. For example for point i and assuming $d_i = 0$, we can predict a previous event at time $t = n\,p$ where $n\,p$ is a period multiple and n is an integer such that:

$$\begin{aligned} C(i, t - \frac{n}{v}) &= v\,(t - \frac{n}{v}) - floor(v\,(t - \frac{n}{v})) \\ &= v\,t - n - floor(v\,t - n) \\ &= v\,t - n - floor(v\,t) + n \\ &= v\,t - floor(v\,t) \\ &= 0 \end{aligned} \tag{5.10}$$

5.4.2 Single Point Tracking

Given a velocity estimate $\hat{v}$ and an observed event e at position $x_1(t)$, we can predict a past event $\hat{e}_p$ triggered by that point at $x_1(t) - n$ and time

$$\hat{t}_p = t - n\,p = t - \frac{n}{\hat{v}}$$

where $p = \frac{1}{\hat{v}}$. In contrast, the true position of the event at $\hat{t}_p$:

$$x_1(\hat{t}_p) = v\,(t - n\,p) = v\,t - \frac{n\,v}{\hat{v}} = x_1(t) - \frac{n\,v}{\hat{v}}$$

Since $\hat{t}_p$ is not necessarily equal to an integer multiple of the sampling period, the past event e_p would have been observed at time

$$t_p = t - n\,p + \eta$$

and location:

$$\begin{aligned} x_1(t - n\,p + \eta) &= v\,t - \frac{n\,v}{\hat{v}} + v\,\eta \\ &= x_1(t) - \frac{n\,v}{\hat{v}} + v\,\eta \end{aligned} \tag{5.11}$$

Substituting this location into the trigger condition (5.9) yields:

$$x_1(t) - \frac{n\,v}{\hat{v}} + v\,\eta - floor(x_1(t) - \frac{n\,v}{\hat{v}} + v\,\eta) = 0$$

Comparing the result to Eq. (5.10) we conclude that $\frac{n\,v}{\hat{v}} - v\,\eta$ is an integer which is equivalent to requiring that:

$$v\,\eta = \frac{n\,v}{\hat{v}} - round(\frac{n\,v}{\hat{v}}) \tag{5.12}$$

Substituting this result into Eq. (5.11), we find that the previously observed event e_p is located at:

$$x_1(t - n\,p + \eta) = x_1(t) - round(\frac{n\,v}{\hat{v}})$$

The euclidean distance between e_p and $\hat{e}_p$ is:

$$\begin{aligned} D_e(e_p, \hat{e}_p) &= x_1(t) - n - (x_1(t) - round(\frac{n\,v}{\hat{v}})) \\ &= round(\frac{n\,v}{\hat{v}}) - n \end{aligned} \tag{5.13}$$

To account for deviation in the past event's position due to the discrete sampling time η, we use Eq. (5.12) to add a time penalty:

$$P(t_p, \hat{t}_p) = v\,\eta = \frac{n\,v}{\hat{v}} - round(\frac{n\,v}{\hat{v}})$$

to the euclidean distance between the events and define a generalized spatiotemporal distance between e_p and $\hat{e}_p$ as:

$$D(e_p, \hat{e}_p) = D_e(e_p, \hat{e}_p) + P(t_p, \hat{t}_p) = \frac{n\,v}{\hat{v}} - n \tag{5.14}$$

We propose to use the generalized distance as an objective function. Figure 5.3 shows an example of the objective function behaviour as a function of velocity for an event moving with a ground truth optical flow of

$$\mathbf{U} = \begin{bmatrix} 1.9 \\ 0 \end{bmatrix}$$

The function reaches a minimum of zero at the ground truth velocity indicating its plausibility.

5.4.3 Multiple Points Tracking

In this section we generalize the results of the previous section to the case where the scene involves k points $\{x_i\}_{i=1}^{k}$. Given an observed event e, a predicted past event $\hat{e}_p$, and a stream of past events $\mathcal{E} = \{e_i\}_{i=1}^{N}$, each point $\{x_i\}_{i=1}^{k}$ will provide a candidate event located at:

$$\begin{aligned} x_i(t - n\,p + \eta_i) &= v\,t + d_i - \frac{n\,v}{\hat{v}} + v\,\eta_i \\ &= x_1(t) + d_i - \frac{n\,v}{\hat{v}} + v\,\eta_i \end{aligned} \tag{5.15}$$

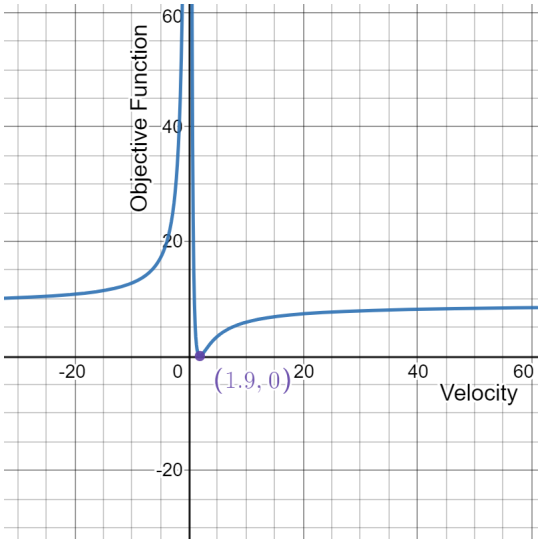


Figure 5.3: Behaviour of the objective function for an event generated by a point moving horizontally with a ground truth velocity of $u = 1.9$.

Substituting into the trigger condition (5.9):

$$x_1(t) + d_i - \frac{n\,v}{\hat{v}} + v\,\eta_i - floor\Big(x_1(t) + d_i - \frac{n\,v}{\hat{v}} + v\,\eta_i\Big) = 0$$

Comparing the result to Eq. (5.10) we conclude that $\frac{n\,v}{\hat{v}} - v\,\eta_i - d_i$ is an integer. This is equivalent to requiring that:

$$v\,\eta_i = \frac{n\,v}{\hat{v}} - d_i - round\Big(\frac{n\,v}{\hat{v}} - d_i\Big) \tag{5.16}$$

Substituting this result into Eq. (5.15), we find that the candidate event e_i are located at:

$$x_i(t - n\,p + \eta_i) = x_1(t) - round\Big(\frac{n\,v}{\hat{v}} - d_i\Big)$$

The euclidean distance between e_i and $\hat{e}_p$ is:

$$\begin{aligned} D_e(e_i, \hat{e}_p) &= x_1(t) - n - \Big(x_1(t) - round\big(\frac{n\,v}{\hat{v}} - d_i\big)\Big) \\ &= round\Big(\frac{n\,v}{\hat{v}} - d_i\Big) - n \end{aligned} \tag{5.17}$$

Using Eq. (5.16) the time penalty can be written as:

$$P(t_i, \hat{t}_p) = \frac{n\,v}{\hat{v}} - d_i - round\Big(\frac{n\,v}{\hat{v}} - d_i\Big)$$

Therefore, the generalized spatiotemporal distance becomes:

$$D(e_i, \hat{e}_p) = D_e(e_i, \hat{e}_p) + P(t_i, \hat{t}_p) = \frac{n\,v}{\hat{v}} - n - d_i$$

and e_p corresponds to the candidate event e_i that satisfies a minimum distance condition with respect to $\hat{e}_p$ such that:

$$L_d(e, \hat{v}) = \min_{\{e_i\}_{i=1}^{k}} \Big(D(e_i, \hat{e}_p)\Big) = \min_{\{e_i\}_{i=1}^{k}} \Big(\frac{n\,v}{\hat{v}} - n - d_i\Big) \tag{5.18}$$

First, we consider the case of a camera moving with a ground truth optical flow of

$$\mathbf{U} = \begin{bmatrix} 1.9 \\ 0 \end{bmatrix}$$

with respect to a scene containing multiple points (for example 3 points) and evaluate the objective function for a single event as a function of velocity u. Figure 5.4 shows the appearance of two spurious minima, marked by red dots, in addition to the minimum, marked by a blue dot, corresponding to the ground truth $u = 1.9$.

Next, we consider the case of processing multiple events $\epsilon = \{e^i\}_{j=1}^{M}$ simultaneously. In this case, we define another objective function for processing M events simultaneously as the sum of the individual generalized distances for each of those events:

$$L_d(\epsilon, \hat{v}) = \sum_{j=1}^{M} \min_{\{e_i\}_{i=1}^{k}} \Big(D(e_i, \hat{e}_p^j)\Big) = \sum_{j=1}^{M} \min_{\{e_i\}_{i=1}^{k}} \Big(\frac{n\,v}{\hat{v}^j} - n - d_i\Big) \tag{5.19}$$

For an example for $M = 3$ and $k = 3$, Figure 5.5(a) shows that the spurious minima, marked with red dots, are larger than the minimum corresponding to the ground truth velocity, marked with a blue dot. The spurious minima only become equal to the ground truth minimum when the initial distances d_i are equal as seen Figure 5.5(b). This case corresponds to a scene with periodic or repetitive patterns.

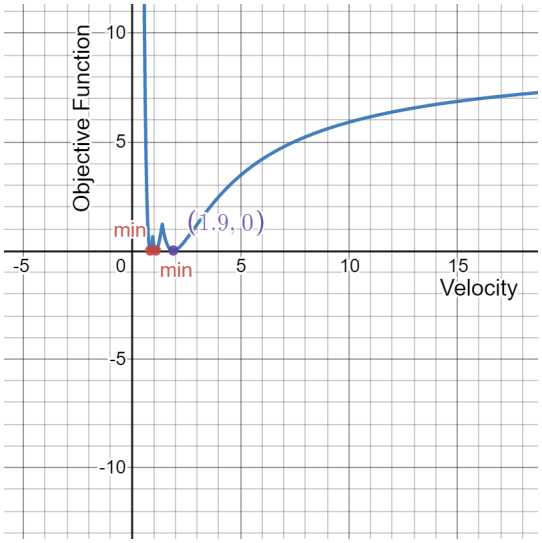
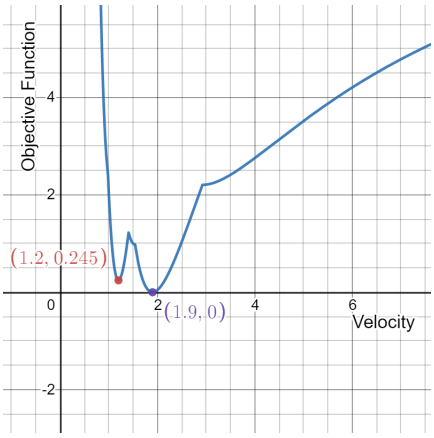


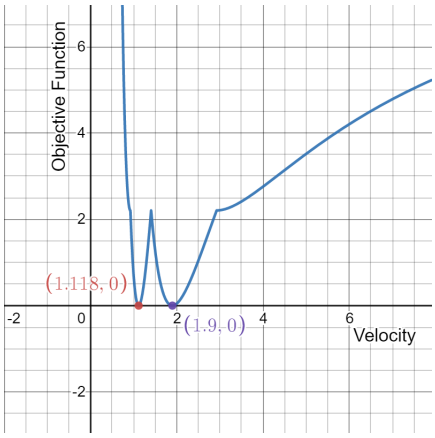
Figure 5.4: Behaviour of the objective function for an observed event in a scene containing 3 points and moving with a ground truth velocity of $u = 1.9$.

5.4.4 Discussion

The mathematical validation showed that an event track belonging to a single camera motion can be used to solve the data association problem and motion estimation. For the special case of a single point moving in 1D motion, We evaluated the effect of the spatial and temporal discretization (quantization) on the prediction of past events. It was then used to suggest an objective function that minimizes a generalized spatiotemporal distance in order to evaluate the optical flow. Test examples showed that scenes containing multiple points and simultaneous processing of events created ambiguity (multiple minima). However, the objective function was able to maintain the minimum corresponding the ground truth even in the presence of spurious minima. We hypothesise that tuning the weight of the time penalty will resolve this ambiguity.



(a) Unequal initial distances



(b) Equal initial distances

Figure 5.5: Behaviour of an objective function defined to process 3 events simultaneously in a scene containing 3 points and moving with a ground truth velocity of $u = 1.9$.

5.5 Experimental Analysis

In this section, we validate experimentally the hypothesis that the consensus of multiple event tracks, where events are processed individually, can be used to solve the data association problem and estimate planar camera motion.

5.5.1 Experimental Setup

We test our hypothesis by varying the time penalty experimentally on a synthetic dataset and an AES generated dataset by tuning the weight r_t of the time penalty in the range of 0 to 25000 pixels/s. In our experiments, we use a spatial window of $n = 3$ and a temporal window made of 50 sampling periods such that $m = 50\mu$s. The maximum generalized distance d_{max} is varied between 3 and 5.

5.5.1.1 Synthetic Dataset

We created an event-based data generator that emulates an idealized AES firing. The simulation provides at each timestep t_i the optical flow of a 32×32 event camera facing a reference image at a known constant depth Z. The ground truth motion is simulated to be constant, to vary linearly, or to vary sinusoidally over time. Grids of 32×32 pixels are sampled from the reference image as the camera moves following the simulated ground truth motion. The events are warped as the camera moves with respect to the reference image. The synthetic data generator is described in details in Appendix B.

5.5.1.2 AES Dataset

We use the state-of-the-art Event-Camera Dataset and Simulator [120] which contains a series of planar scenes collected via the Davis AES [22]. The spatial resolution of this AES is 240×180 pixels, its sampling period is $\delta t = 1\mu$s, and its dynamic range is (130 dB).

We use all planar motion sequences available in the dataset, namely *slider close*, *slider far*, *slider hdr close*, and *slider hdr far*. They were recorded by mounting the AES to an automated linear slider moving parallel to a wall at a constant depth. Each sequence lasts about 5 seconds and contains 4–10 million events. The first of the first and third sequences is small whereas that of the second and fourth sequences is large. The latter two sequences feature a high-dynamic-range (HDR) created using spotlights.

5.5.1.3 Metrics

To evaluate the performance of our approach, we compute the average end-point error (*AEE*) of the optical flow. It measures the 'distance' between the

endpoints of the predicted and ground truth translational camera motions:

$$AEE = \frac{1}{N_e} \sum_{i=1}^{N_e} \sqrt{(u_{i,pred} - u_{i,g})^2 + (v_{i,pred} - v_{i,g})^2} \tag{5.20}$$

where N_e is the number of events evaluated. We also calculate the relative AEE which normalizes the AEE with respect to the ground truth to obtain the relative AEE:

$$AEE_{rel} = \frac{1}{N_e} \sum_{i=1}^{N_e} \frac{\sqrt{(u_{i,pred} - u_{i,g})^2 + (v_{i,pred} - v_{i,g})^2}}{\|\mathbf{U}_{i,g}\|} \tag{5.21}$$

where $\|\mathbf{U}_{i,g}\|$ is the norm of the ground truth motion at timestep t_i

$$\mathbf{U}_{i,g} = \begin{bmatrix} u_{i,g} \\ v_{i,g} \end{bmatrix}$$

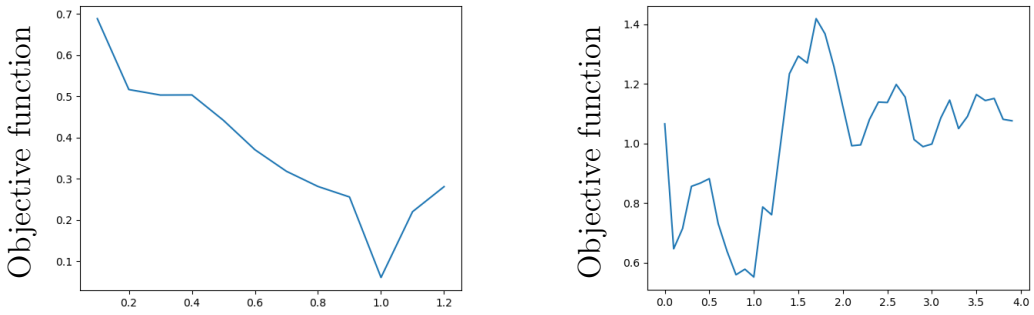
5.5.2 Demonstration of the Objective Function

First, we examine the feasibility of tuning the time penalty to eliminate ambiguities in the objective function. Specifically, we evaluate the objective function for a simple case (Case I) representing a camera moving with a ground truth optical flow of

$$\mathbf{U} = \begin{bmatrix} 0.8 \\ 0 \end{bmatrix}$$

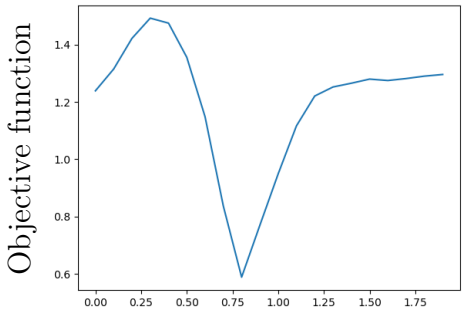
In each experiment, we consider a set of camera motions distributed uniformly to cover the two dimensional OF space ranging from 0 to 10 pixels/timestep in both directions. We evaluate the objective function for each value of OF using different values of r_t. The results for 4 selected values of r_t are shown in Fig. 5.6.

We found that setting the time penalty weight to smaller values, such as $r_t = 0$ in Fig. 5.6(a) or $r_t = 2000$ in Fig. 5.6(b), results in the function reaching an incorrect minimum away from the ground truth motion, multiple

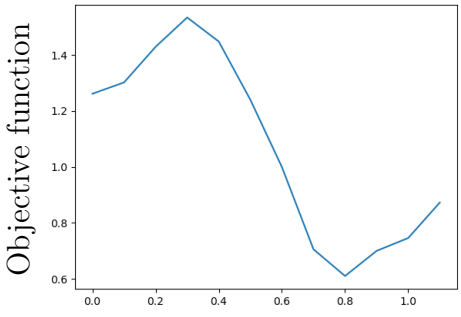


(a) $r_t = 0$

(b) $r_t = 2000$



(c) $r_t = 10000$



(d) $r_t = 20000$

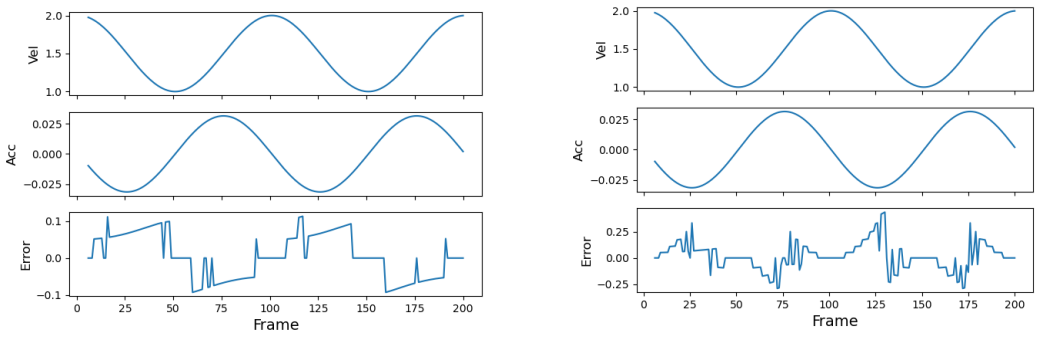
Figure 5.6: Effect of time penalty term r_t on the objective function for a ground truth velocity $u = 0.8$.

ambiguous minima or both. On the other hand, setting r_t to larger values, such as $r_t = 10000$ in Fig. 5.6(c) or $r_t = 20000$ in Fig. 5.6(d), results in the function reaching a global minimum at the ground truth velocity. This shows that ablating ($r_t = 0$) or undertuning the time penalty results in an erroneous data association due to the ambiguity caused by uncorrelated events. Varying the ground truth velocity or maximum distance d_{max} did not change these results.

To showcase the effect of undertuning the time penalty, we consider a

camera moving with a sinusoidal ground truth velocity varying from 1 to 2 pixel/timestep over a period of 100 timesteps (frames). For each experiment, we evaluate the objective function following the procedure described above. Inspecting the resulting objective function, the predicted OF $\mathbf{U}_{i,pred}$ is found as that satisfying the criterion 5.3 and used to evaluate AEE.

Figs. 5.7(a) and (b) compare the AEE obtained for two values of the time penalty weight for $r_t = 20000$ and $r_t = 2000$, respectively. The larger r_t value results in a 4 fold increase in accuracy for planar motion estimation. This highlights the importance of the time penalty weight.



(a) $r_t = 20000$ (b) $r_t = 2000$

Figure 5.7: Comparison of the AEE over a sequence of 200 synthetic frames for the same sinusoidal ground truth velocity and acceleration and two values of the time penalty weight r_t.

Next, we use the synthetic dataset to demonstrate the feasibility of the distance-based method on a more complex camera motion such as (Case II):

$$\mathbf{U} = \begin{bmatrix} 0.8 \\ 0.6 \end{bmatrix}$$

and evaluate the generalized distance as a function of OF over the two dimensional grid described above. For a time penalty weight $r_t = 10000$, Fig. 5.8 shows that the global minimum, marked by a red dot, corresponds to the ground truth motion.

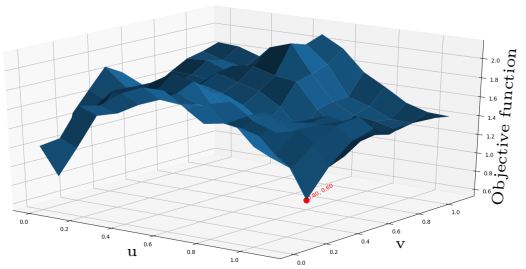


Figure 5.8: The objective function evaluated over an OF range for a ground truth OF of $(u, v) = (0.8, 0.6)$.

5.5.3 Results

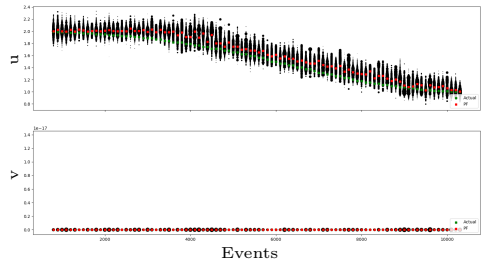
In this section, we demonstrate the ability of our approach to estimate planar motion. We use our distance-based method to solve the data association problem. Instead of the brute force uniform discretization approach employed above, we exploit our particle filter framework to search the optical flow space for estimating planar motion.

Initially, all particles are uniformly initialized to cover an OF space ranging from 0 to 10 pixels/timestep in both directions, and the number of particles N is varied between 100 and 400. Particles inconsistent with the measured event e receive a generalized distance of d_{max} and, therefore, a low likelihood score that updates its weight via Eq. (5.7).

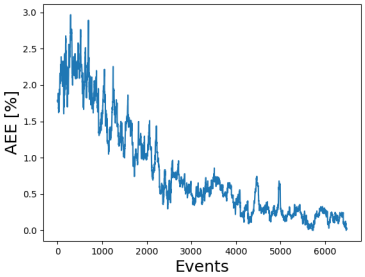
5.5.3.1 Synthetic Dataset

We consider a camera moving with 3 different types of ground truth motions, namely constant, linearly varying, and sinusoidally varying velocities. For every 100th incoming event, we show in Figs. 5.9(a), (c) and (e) the ground truth velocity (green dots), the predicted OF (red dots), and the

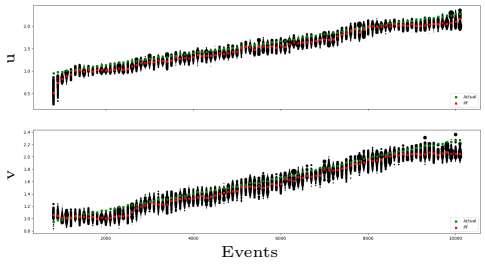
particle distribution (black dots). The results show qualitatively that our approach can accurately track the ground truth motion in all 3 cases. The corresponding relative AEE, shown in Figs. 5.9(b), (d) and (f), confirm this conclusion quantitatively. Our method accurately estimates various types of planar camera motion with average errors less than 1%.



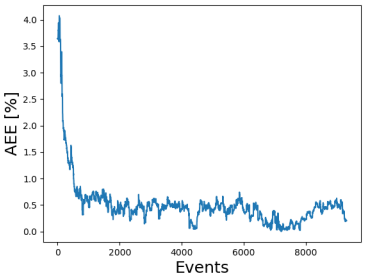
(a) Sinusoidal velocity



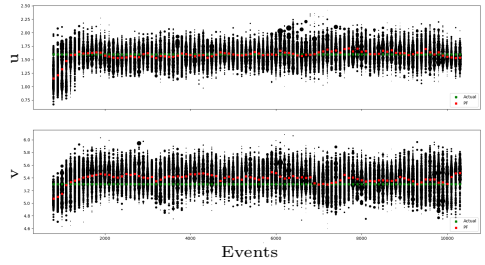
(b) Sinusoidal velocity error



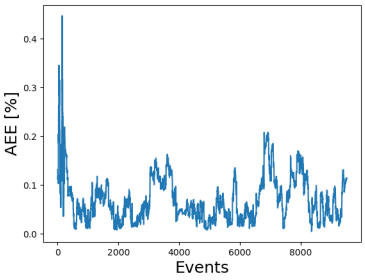
(c) Linear velocity



(d) Linear velocity error



(e) Constant velocity

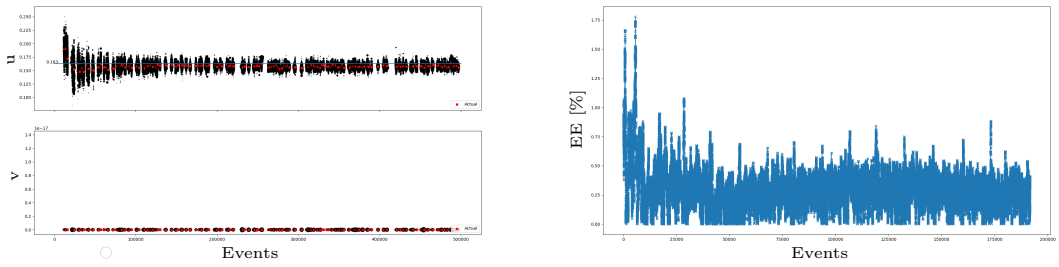


(f) Constant velocity error

Figure 5.9: Comparison of (a) sinusoidal, (c) linear and (e) constant ground truth motions (green), our predictions (red), particle distributions (black), and (b), (d) and (e) their corresponding relative AEE, respectively.

5.5.3.2 AES Dataset

As a sample case, we show in Fig. 5.10(a) the ground truth velocity (blue dots), the predicted OF (red dots), and the particle distribution (black dots) for a sample of $200,000$ events from the *slider far* sequence. The results show qualitatively that our approach can accurately track the ground truth motion on an AES sequence. Figure 5.10(b) shows the corresponding Endpoint Error (EE) per event. It confirms our conclusion quantitatively. Our method accurately estimates the planar camera motion with an average error less than 0.6% over the sample. Similar results were obtained for all four motion sequences. This performance is attributed to our asynchronous distance-based method capability to correctly associate events and their previous firings along event tracks.



(a) *Slider far* velocities (b) *Slider far* error

Figure 5.10: Sample performance of our approach on the *slider far* sequence. (a) It accurately tracks the planar OF vs a sample of 2×10^5 events. (b) Relative Endpoint Error (EE) vs a sample of 2×10^5 events for the *slider far* sequence.

We compare in Table 5.1 the performance of our approach to those of the event-based SAE approach [18] and the MATLAB implementation of the frame-based LK approach [106]. The latter approach was implemented on the companion grayscale images provided in the dataset. We used the code

provided by Benosman et.al [18] to evaluate the camera motions. The SAE approach [18] accumulates event within a few millisecond windows into an image frame to track the camera motion. We used windows of 10 ms in our implementation of SAE. Eq. (5.21) was used to evaluate the AEE_{rel} in all sequences.

Table 5.1: Comparison among the asynchronous planar motion approach, the event-based SAE [18], and the frame-based LK [106] in terms of the Relative Average Endpoint Errors (AEE_{rel}).

Approaches	slider far	slider close	slider hdr far	slider hdr close
	AEE_{rel}[%]	AEE_{rel}[%]	AEE_{rel}[%]	AEE_{rel}[%]
This Approach	**0.95**	**1.1**	**0.97**	**1.08**
SAE [18]	3.9	3.7	4.9	4.6
Frame-based LK [106]	14.7	13.9	-	-

Our approach clearly outperforms both the SAE and LK approaches on all 4 sequences. In particular, while LK approach fails on the high dynamic range (hdr) sequences, due to the degradation in the quality of the grayscale images affected by the large intensity differences in those scenes. Our approach maintains the same accuracy level. This shows the advantages of events cameras in hdr situations.

5.5.4 Discussion

5.5.4.1 Initialization of the Particle Filter

Carrying out an effective search for precedents of an event requires a history of previous events. As a result, starting the particle filter search algorithm requires an initialization period during which this history is generated. We show an example of this initialization period by reporting the EE for every 100th incoming event over the first 10000 events of the *slider far* sequence

in Fig. 5.11. The figure shows high errors during the first few thousands events which diminishes reaching less than 2.5% EE beyond 3000 events as the history of events acquires the necessary amount. This behaviour was observed for all other sequences as the synthetic dataset, Appendix C. We recommend that the initialization period be omitted from the motion tracking results.

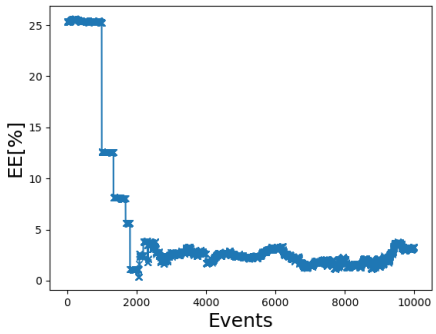


Figure 5.11: Relative Endpoint Error (EE) per 100th event for the first 10,000 events of the *slider far* sequence.

5.5.4.2 Rapid Motion Limitations

In this section, we test the limits at which the velocity constancy assumption starts to undermine the accuracy of our motion estimation approach. Towards that, we consider a synthetic dataset generated by our simulator described above where the camera is moving with a ground truth horizontal velocity that varies sinusoidally around a mean of 1.5 pixel/timestep over a period of 100 timesteps (frames)

$$\mathbf{U} = \begin{bmatrix} 1.5 + p\cos(\frac{2\pi}{100}t) \\ 0 \end{bmatrix}$$

As the amplitude of the velocity variation p is increased, we evaluate the objective function for an OF range from 0 to 10 pixels/timestep, determine

the predicted OF $\mathbf{U}_{i,pred}$ as that satisfying the criterion 5.3 and used it to evaluate AEE.

Fig. 5.12 shows samples of the AEE for four values of the sinusoidal peak to peak velocity variations and their corresponding accelerations. Comparing the error profiles shown in Fig. 5.12, we found that unlike the velocity, which varies periodically, the AEE was not periodic. In other words, the errors encountered are not directly proportional to the magnitude of velocity. This is expected since even where velocity variation is periodic the scene features, and therefore event tracks, are not necessarily periodic. However, similar velocity magnitudes had similar levels of AEE regardless of the terminal peak to peak velocities. As the peak-to-peak velocity variation increased the AEE increased monotonically signaling a loss in accuracy in Figs. 5.12(c) and (d). However, the error was in all cases bounded and diminished as the velocity approached its mean value.

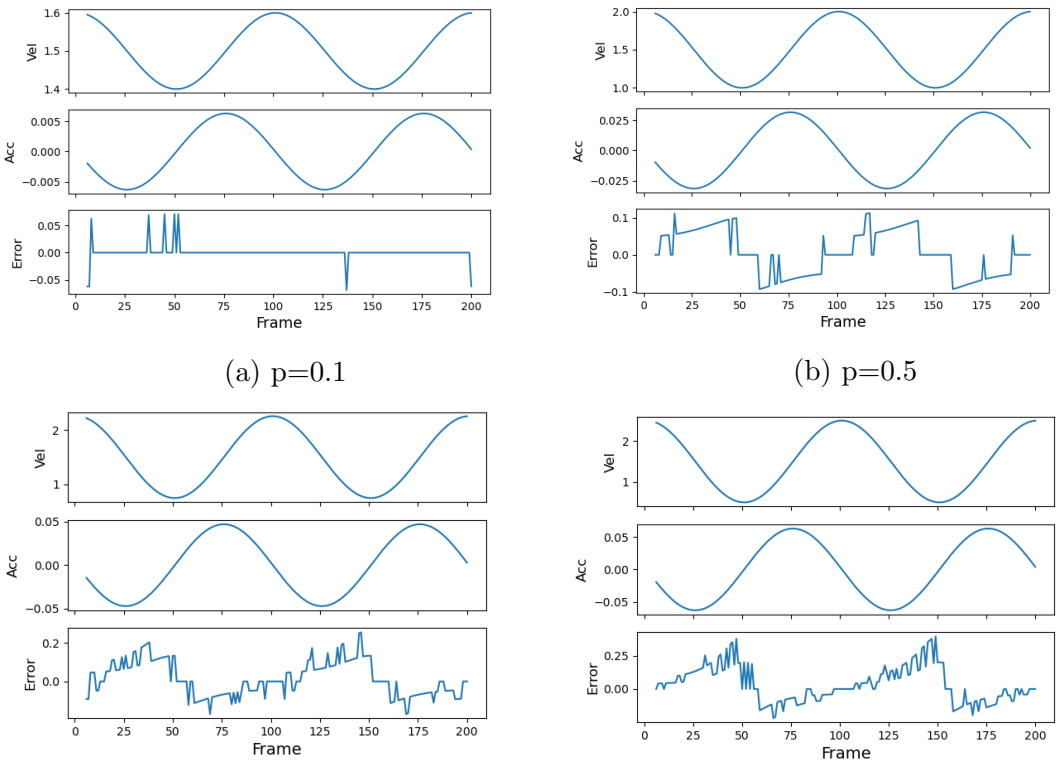
To examine more closely the dependence of the AEE on the magnitude of velocity, and the limits of our approach's accuracy, we consider another synthetic dataset with a constant ground truth velocity of

$$\mathbf{U} = \begin{bmatrix} u \\ 0 \end{bmatrix}$$

For each value of u, we evaluated the AEE over 200 timesteps of the camera motion. Five trials were conducted starting from five different points in the reference image. The average AEE was calculated over all trials. This procedure was repeated for the velocity u range from 0.1 to 12 pixels/timestep in steps of 0.1 pixels/timestep. The results are shown in Fig. 5.13. The error sensitive to velocity increases significantly beyond 3 pixels/timestep.

5.5.4.3 Texture Limitations

Textured scenes trigger event cameras to produce denser event streams (more events per second). To visualize this effect, we show in Fig 5.14 50 event



(a) p=0.1

(b) p=0.5

(c) p=0.75

(d) p=1

Figure 5.12: Samples of the AEE for a camera moving with 4 selected sinusoidally varying velocities as a function of the timestep (frame).

frames in a sparse scene (top) obtained by setting the data simulator to low sensitivity and a dense scene (bottom) obtained by increasing the simulator's sensitivity over the same area of the reference image. The dense scene quadruple the number events compared to the sparse scene, filling on average more than 70% of the image area (70% of all available 32×32 pixels) compared to 20% of the area in the sparse scene.

To quantify the impact of textured scenes on our approach, we consider

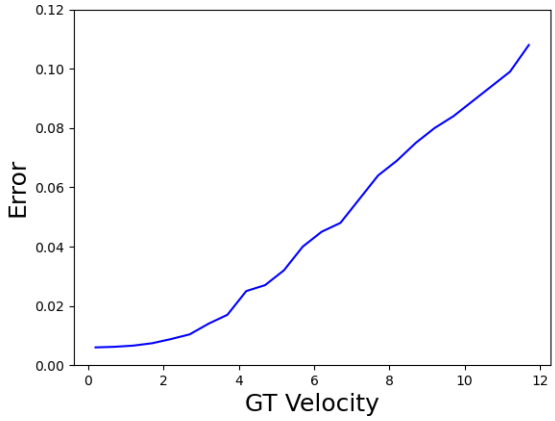


Figure 5.13: The mean AEE as a function of a constant ground truth velocity.

a camera moving with a ground truth velocity of

$$\mathbf{U} = \begin{bmatrix} 0.8 \\ 0 \end{bmatrix}$$

We vary the sensitivity of our synthetic data generator to increase the observed texture of the reference image. For each value of the sensitivity, we evaluate the AEE as the camera moves for 1 second with respect to the reference image and calculate the mean AEE for all events. We repeat this evaluation as the scene texture is varied in 30 steps and show in Fig. 5.15 the mean AEE as a function of the texture expressed in terms of the total number of events generated during the motion. The results show that the accuracy of the motion estimation decreases significantly with higher texture. For instance, the AEE for more textured scene that fired about 35,000 events/s (filling 70% of the area) is six times more than that of the sparse scene generating 10,000 events/s (filling 20% of the area). This behaviour is explained by the fact that events in textured scenes fire very closely in space and time, making the problem of data association harder to resolve.

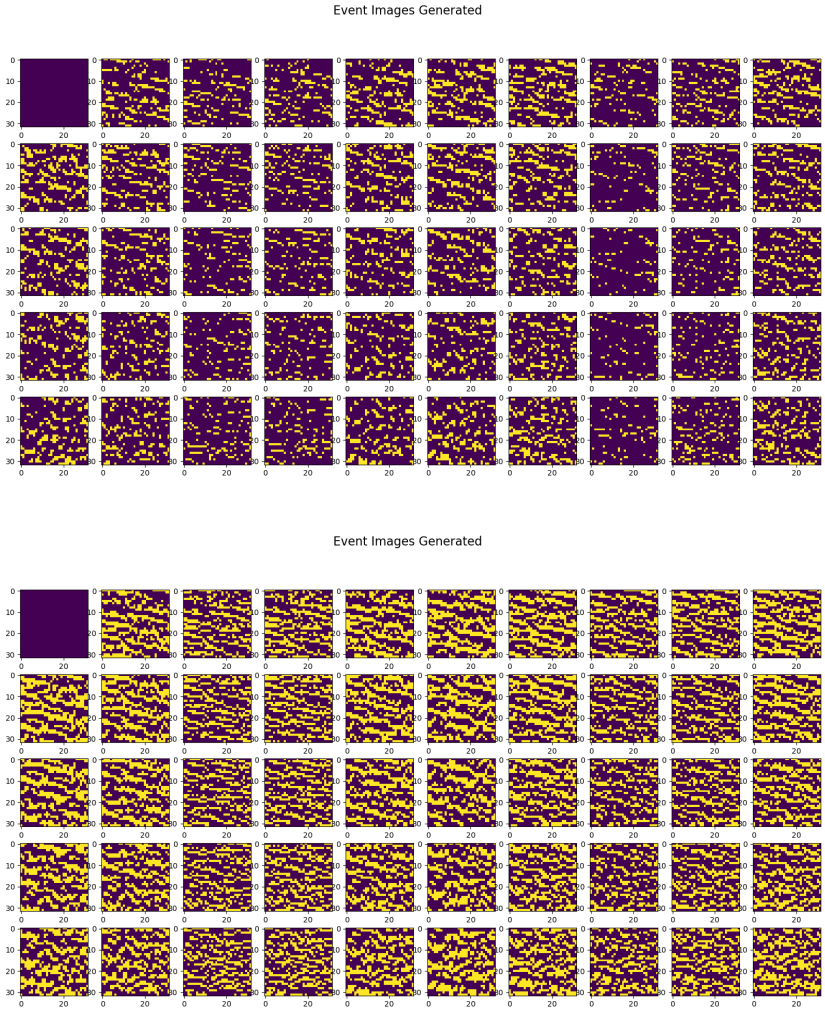


Figure 5.14: Generated event frames for a true ground truth velocity $u = 0.8$, Left: 20% of area is filled with events, and right: 70% of area is filled with events

5.6 Conclusion

In this chapter, we have validated the thesis hypothesis by showing that our proposed method which handles each event individually without assuming

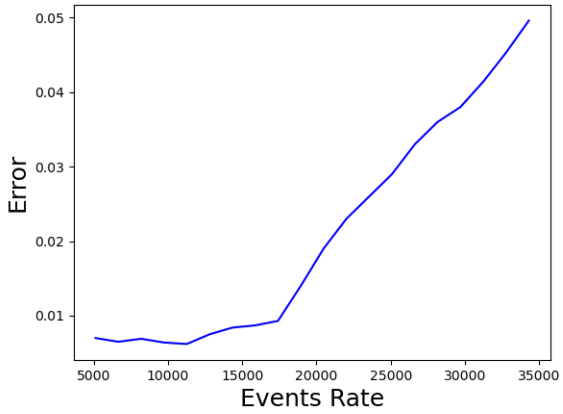


Figure 5.15: The mean AEE as a function of scene texture for a camera moving with a ground truth motion $u = 0.8$.

any relationship to its neighbors beyond the consistency with the camera motion, is sufficient for correct data association of events and their previous firing along corresponding tracks. Our results show that this can successfully identify the correct motion in situations of high speed, high dynamic range and strong illumination changes. We characterized cases that limit its accuracy, mainly in highly textured environments and very fast motion. Further, our results on real data are an order of magnitude more accurate than optical flow based approaches.

Chapter 6

Asynchronous General Motion Estimation

6.1 Introduction

In this chapter, we extend the voting based approach presented in the previous chapter to solve the general form of the event-based motion estimation problem Eqs. (3.4) and (3.6). In fact, our extended approach is also valid for the case of general camera motion and arbitrary scene depth.

In the case where the camera is undergoing an arbitrary motion, each pixel may have its own depth. To address that, we create a new asynchronous voting based method, which predicts forward a set of candidate image velocities from the data association, then perform a search over all those candidates to vote for the one that is most consistent with the camera motion. This allows us to formulate a new depth-less objective function.

6.2 Method

The general motion equation, Eq. (3.4), relates the image velocities (Optical flow) $\mathbf{U}(\mathbf{x})$ and the camera translational and angular velocities $(\mathbf{\Omega}, \mathbf{T})$ by:

$$\mathbf{U(x)} = \frac{1}{Z(\mathbf{x})} A(\mathbf{x})\mathbf{T} + B(\mathbf{x})\mathbf{\Omega} \tag{6.1}$$

where

$$A(\mathbf{x}) = \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix} \quad , \quad B(\mathbf{x}) = \begin{bmatrix} \frac{xy}{f} & \frac{-f+x^2}{f} & y \\ \frac{f+y^2}{f} & \frac{-xy}{f} & -x \end{bmatrix}$$

Given a camera velocity $(\mathbf{\Omega}, \mathbf{T})$ and an image velocity $\mathbf{U}$, we want to quantify the consistency of the image velocity, obtained from the data association (Eq. (3.6)), with the camera velocity. While it is possible to solve for the unknown depth $Z(\mathbf{x})$ based on smoothness, by assuming that pixels that fire within a close spatiotemporal distance represent areas that have the same depth. However, as per our problem formulation we want to avoid the use of smoothness assumptions. We, therefore, present a new voting-based method to estimate general motion by solving for the unknown depth.

The distance-based method outlined in Section 5.2 involved a backward projection $\hat{e}_p$ of an event $e(\mathbf{x}, t, p)$ to associate it to its previous firing e_p. We now propose a different way to solve the data association problem, (Eq. (3.6)), by searching for all previously fired events $\mathcal{E}_c$ in a spatiotemporal window around $\mathbf{x}$, evaluating candidate image velocities $\mathbf{U^c}$ that forward project them into e then minimizing an objective function to determine the candidate flow consistent with the camera motion.

Let $\mathcal{E}_c(\mathbf{x}, n, m) \in \mathcal{E}$, colored dots in Fig. 6.1, represent the set of candidate events within an $n \times n$ spatial window centered around the current event location $\mathbf{x}$, and a temporal window $\pm m\,\delta t$ centered around $t - q\,\delta t$ the previous firing time at $\mathbf{x}$ where $m \leq q$. Using Eq. (3.6) we project each candidate event e_c forward to obtain a candidate image velocity:

$$\mathbf{U^c} = \left(\frac{\mathbf{x} - \mathbf{x_c}}{t - t_c} \right) \tag{6.2}$$

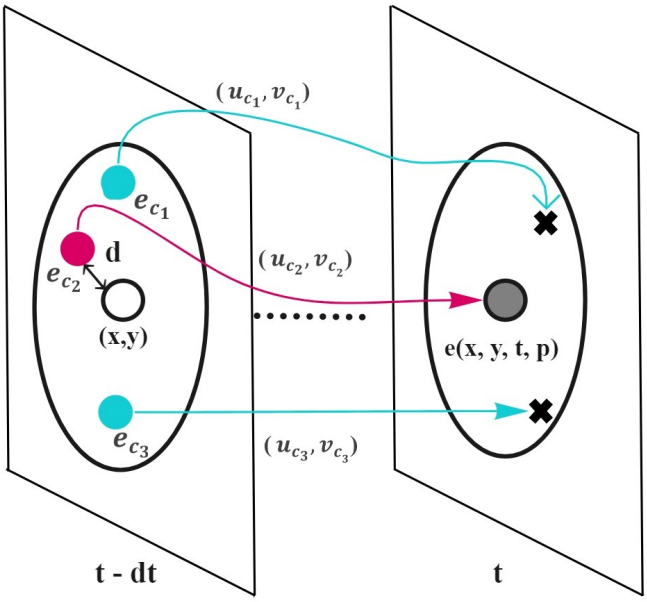


Figure 6.1: Graphical outline of the forward prediction method. For an input event e, look back at this pixel $\mathbf{x}$ just before this event fired and search for previous event candidates in a spatiotemporal window around $\mathbf{x}$. Project each candidate forward to predict the image velocity $\mathbf{U^c}$ corresponding to the correct e_p that fired e (red in this example).

Only the correct flow candidate $\mathbf{U^c}$ that is consistent with the camera motion $(\mathbf{T}, \mathbf{\Omega})$ will associate the correct prior event e_p to e.

Given $\mathbf{U}^c$, three groups of unknowns are left in Eq. (6.1): the camera translational and angular velocities $\mathbf{T}$ and $\mathbf{\Omega}$ and the depth $Z(\mathbf{x})$. Bruss and Horn [27] derived a constraint that eliminates the depth from Eq. (6.1) by resetting it in the form:

$$\mathbf{T}^t \cdot (\mathbf{x} \times \mathbf{U}) + (\mathbf{T} \times \mathbf{x})^t \cdot (\mathbf{x} \times \mathbf{\Omega}) = 0 \tag{6.3}$$

which reduces to:

$$\begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix}^t \Big(\underbrace{\begin{pmatrix} fv \\ -fu \\ yu - xv \end{pmatrix}}_{[\mathrm{I}]} - \begin{pmatrix} -(f^2+y^2) & xy & fx \\ xy & -(f^2+x^2) & fy \\ fx & fy & -(x^2+y^2) \end{pmatrix} \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \Big) = 0 \tag{6.4}$$

where the matrix $[I]$ is a function of the image velocity. We used Kanatani's [83] normalized translational form:

$$\boldsymbol{\tau}(\mathbf{x}, \mathbf{T}) = \frac{1}{\|A(\mathbf{x})\mathbf{T}\|} \begin{pmatrix} \Big[A(\mathbf{x})\mathbf{T}\Big]_y \\ -\Big[A(\mathbf{x})\mathbf{T}\Big]_x \end{pmatrix} \tag{6.5}$$

where $[V]_x$ and $[V]_y$ are the x and y components of a 2D vector $\mathbf{V}$, to rewrite the bilinear constraint in the form:

$$(\mathbf{U} - B \cdot \boldsymbol{\Omega})^t \big(\|A \cdot \mathbf{T}\| \boldsymbol{\tau}\big) = 0 \tag{6.6}$$

$$F(\mathbf{U}, \mathbf{T}, \boldsymbol{\Omega}) = 0 \tag{6.7}$$

where F is the bilinear constraint.

The scale ambiguity [9], imposed by the appearance of the translational velocity and depth as a ratio $\frac{\mathbf{T}}{Z}$ in Eq. (6.1), means that we can only infer the direction of translation (the unit vector $\hat{\mathbf{T}}$) known as the translational heading, but not its magnitude. Expressing the translational heading in spherical coordinates, it can be written as:

$$\hat{\mathbf{T}}(\phi, \theta) = \begin{pmatrix} \sin\theta\cos\phi \\ \sin\theta\sin\phi \\ \cos\theta \end{pmatrix} \tag{6.8}$$

where θ is the polar angle between the x-axis and the polar axis, and ϕ is the azimuthal angle between the z-axis and the translational $\hat{\mathbf{T}}$ unit vector.

Given a set of candidates image velocities $\mathcal{E}_U = \{\mathbf{U_i^c}\}_{\mathbf{i=1}}^{\mathbf{k}}$ for a current event e, we define a generalized motion function as:

$$G(\mathbf{U_i^c}, \hat{\mathbf{T}}, \boldsymbol{\Omega}, t_c, q) = \begin{cases} \min(F(\mathbf{U_i^c}, \hat{\mathbf{T}}, \boldsymbol{\Omega}) + P(t_c, q), V_{max}); & p_i = p \\ V_{max}; & \text{otherwise} \end{cases} \tag{6.9}$$

The function $P(\mathbf{t})$ is a time penalty function:

$$P(t_c, q) = r_t |t_c - (t - q\delta t)| \tag{6.10}$$

where r_t is the time penalty weight and V_{max} is a maximum motion penalty.

In case of pure rotational motion, the translational velocity is $\mathbf{T} \approx 0$ rendering the bilinear constraint singular. Under this condition, the motion equation Eq. (6.1) reduces to:

$$\begin{aligned} u &= f\omega_y - y\omega_z + \frac{x^2}{f}\omega_y - \frac{xy}{f}\omega_x \\ v &= x\omega_z - f\omega_x - \frac{y^2}{f}\omega_x + \frac{xy}{f}\omega_y \end{aligned} \tag{6.11}$$

and we replace the function F by a rotational function F_{rot} derived directly from the above equation:

$$F_{rot}(\mathbf{U^c}, \boldsymbol{\Omega}) = ||\mathbf{U}^c - \mathbf{U}|| = (u^c - u_{rot})^2 + (v^c - v_{rot})^2 = 0 \tag{6.12}$$

Using the generalized motion function, we define an objective function for general motion that votes the minimum camera motion over all image velocity candidates:

$$M(e, \hat{\mathbf{T}}, \boldsymbol{\Omega}) = \min_{\mathbf{U_i^c} \in \mathcal{E}_U} (G(\mathbf{U_i^c}, \hat{\mathbf{T}}, \boldsymbol{\Omega}, t_c, q)) \tag{6.13}$$

6.3 Framework

We extend our planar motion particle filter, Section 5.3, to use the objective function defined above as the likelihood that votes the image velocity consistent with the camera motion. The framework is described schematically in Figure 6.2 and procedurally in Algorithm 2.

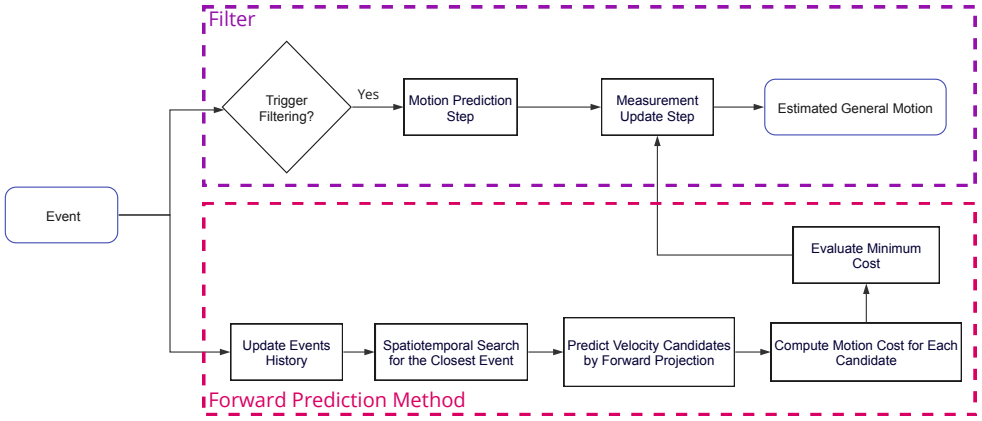


Figure 6.2: A schematic of the framework. For a current event e, our forward prediction method serves as the likelihood in a particle filter framework to vote the flow candidate which is most consistent with the camera motion.

For a current event e, we identify a set of forward flow candidates $\mathcal{E}_U$. The filter is represented by a set particles, $P^{(t)} = \{P_j^{(t)}\}_{j=1}^{N}$ for each flow candidate $\mathbf{U}_i^c$. Each particle consists of the current state $(\hat{\mathbf{T}}_j^{(t)}, \mathbf{\Omega}_j^{(t)})$ and its weight $w_j^{(t)}$. We update the state of every particle via the motion prediction step such that:

$$\hat{\mathbf{T}}_j^{(t)} = \hat{\mathbf{T}}_j^{(t-\tau)} + \mathcal{M}(\hat{\mathbf{T}}_j^{(t)}) \tag{6.14}$$

$$\mathbf{\Omega}_j^{(t)} = \mathbf{\Omega}_j^{(t-\tau)} + \mathcal{M}(\mathbf{\Omega}_j^{(t)}) \tag{6.15}$$

where $(\hat{\mathbf{T}}_j^{(t-\tau)}, \mathbf{\Omega}_j^{(t-\tau)})$ is the previous state of particle j at $t-\tau$, $\mathcal{M}(\hat{\mathbf{T}}_j^{(t)})$ and $\mathcal{M}(\mathbf{\Omega}_j^{(t)})$ are sampled from a Gaussian distribution independently for each vector such that $\mathcal{M}(\hat{\mathbf{T}}_j^{(t)})$ and $\mathcal{M}(\mathbf{\Omega}_j^{(t)}) \sim N(0, \sigma^2)$, and σ is the standard deviation.

In the measurement update step, we modify Eq. (5.7) to calculate the likelihood $P(e \mid \hat{\mathbf{T}}_j^{(t)}, \hat{\mathbf{\Omega}}_j^{(t)})$ for each particle given the measured event e as:

$$P(e \mid \hat{\mathbf{T}}_j^{(t)}, \mathbf{\Omega}_j^{(t)}) = \exp(-\alpha_p M_j^{(t)}(e)) \tag{6.16}$$

where α_p is a scaling decay parameter. After assigning a likelihood to each particle, we normalize the distribution, and resample as described in Section 5.3. The camera egomotion $(\hat{\mathbf{T}}_j^{(t)}, \boldsymbol{\Omega}_j^{(t)})$ is estimated as the weighted average over all the particles.

As shown in Fig. 6.2 and algorithm 2, for a current event e, a single predicted flow candidate votes multiple camera motions (particles), in order to find the best set of image and camera velocities that minimizes the objective function Eq. (6.13). This is a more efficient approach than that adopted in the planar motion framework.

Algorithm 2: Asynchronous General Motion Estimation

Input: Current Event

Output: Camera Egomotion (Heading and Angular Velocities)

Initialize $N \times 5$ particles (camera velocities) with uniform weights;
if $p > 0$ **then**
    Add and update the positive polarity history;
else
    Add and update the negative polarity history;
end
if *enough events* **then**
    Motion prediction step using Eq. 6.15;
    Search for the closest events within a spatiotemporal window of e;
    Predict forward flow candidates using Eq. 6.2;
    for *each particle in* N **do**
        if $\boldsymbol{T} \approx 0$ **then**
            Evaluate F for each candidate using Eq. 6.12;
        else
            Evaluate F for each candidate using Eq. 6.7;
        end
        Compute each candidate's motion cost using Eq. 6.9;
        Evaluate the total cost using Eq. 6.13;
        Compute the likelihood from the returned cost using Eq. 6.15;
        Update the particle's weight using Eq. 5.8;
    end
    Normalize the weights of all particles;
    Resample if $N_{eff} \leq \frac{N}{2}$;
    Output the mean camera heading and angular velocities of all particles;
end

6.4 Experimental Setup

To experimentally validate our hypothesis that multiple event tracks can be used to solve the data association problem, we use a real dataset and process each event individually. The weight r_t of the time penalty is varied in the range of 0 to 25000 pixels/s. In our experiments, we use a spatial window of $n = 4$ and a temporal window made of $m = 50$ resulting in a temporal window of $\pm 50\mu$s. The maximum motion cost V_{max} is varied between 2 and 3. All particles are uniformly initialized to cover angular velocities ranging from -20 to 20 rad/s and all 360 degrees translational headings. The number of particles N is varied between 500 and 1500. Particles inconsistent with the measured event e receive a maximum motion cost of V_{max} and, therefore, a low likelihood score that updates its weight via Eq. (6.16)

6.4.1 Egomotion Dataset

We use a state-of-the-art Event-Camera Dataset and Simulator [120] which contains scenes collected via the Davis [22] AES with a spatial resolution of 240×180 pixels, a sampling period of $\delta t = 1\mu$s, and a dynamic range 130 dB. The Davis also has a built in Inertial Measurement Unit (IMU), which measures the rates of the tilt (pitch), pan (yaw) and roll (optical axis rotation) around the X, Y, and Z axes of the camera, Figure 6.3.

The dataset includes the 2D translation sequences described in Section 5.5.1.2 as well as general motion sequences that we will use in this section. Specifically, we use two rotational sequences, namely *shapes rotation* and *boxes rotation*, where the latter features a highly textured scene, and four general motion sequences, namely *shapes translation*, *shapes 6dof*, *poster 6dof*, and *hdr poster*, where the latter sequence features a high-dynamic-range (HDR) created using spotlights. These sequences contain ground truth velocities from the IMU collected at 1KHz, and ground truth camera pose from a motion capture system (mocap) collected at 200 Hz. Each sequence is about 1

minute long and contains 100–200 million events.

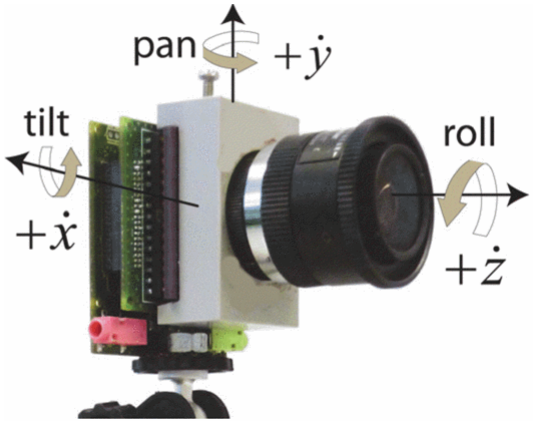


Figure 6.3: Definition of the Davis IMU reference frame and axes definition [47].

6.4.2 Ground Truth

We recover the ground truth linear velocity and angular velocity of the camera. Let $[\mathbf{P}]$ be the 4×4 camera pose at time t:

$$\mathbf{P} = \begin{bmatrix} \mathbf{R} & \mathbf{X} \\ 0 & 1 \end{bmatrix} \tag{6.17}$$

where $\mathbf{X}$ is a 1×3 translation vector and $[\mathbf{R}]$ is a 3×3 rotation matrix such that $[R] \in SO(3)$. Given $[\mathbf{P}_1]$ and $[\mathbf{P}_2]$, the camera poses at t_1 and t_2 respectively, the transformation matrix $[\mathbf{M}_{21}]$ from t_1 to t_2 is:

$$\mathbf{M}_{21} = \mathbf{P}_2^{-1}\mathbf{P}_1 = \begin{bmatrix} \mathbf{R}_{21} & \mathbf{X}_{21} \\ 0 & 1 \end{bmatrix} \tag{6.18}$$

where $[\mathbf{R}_{21}]$ is the resulting 3×3 relative rotation matrix and $\mathbf{X}_{21}$ is the resulting relative translation vector.

Assuming constant acceleration, the ground truth translational velocity $\mathbf{T}_g$ and angular velocity $\mathbf{\Omega}_g$ of the camera can be obtained by numerical differentiation as:

$$\mathbf{T}_g = \frac{\mathbf{X}_{21}}{t_2 - t_1} \tag{6.19}$$

$$\breve{\mathbf{\Omega}} = \frac{logm(\mathbf{R}_{21})}{t_2 - t_1} \tag{6.20}$$

where $logm$ is the logarithmic map from $SO(3)$ to $so(3)$. It converts the rate of $[\mathbf{R_{21}}]$ into the corresponding skew symmetric matrix:

$$\breve{\mathbf{\Omega}} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{6.21}$$

Since we can only infer the camera heading $\hat{\mathbf{T}}$ not the actual translational velocities $\mathbf{T}$, we normalize the ground truth translation velocity vector to obtain $\hat{\mathbf{T}}_g$ and recover the ground truth heading ϕ and θ using the spherical coordinates equations (6.8).

6.4.3 Performance Metrics

To evaluate the performance of our approach, we compute the Axis (orientation) Error for the translational heading ($AE_{\hat{T}}$) and the angular velocity (AE_{Ω}). For a timestep t_i, they measure the distance between the sampled ground truth of the camera motion and that predicted using the closest event to it. They are defined as:

$$AE_{\hat{T}}(t_i) = \cos^{-1}(\hat{\mathbf{T}}_{pred} \cdot \hat{\mathbf{T}}_g) \tag{6.22}$$

$$AE_{\Omega}(t_i) = \cos^{-1}(\mathbf{\Omega}_{pred} \cdot \mathbf{\Omega}_g) \tag{6.23}$$

We also compute their Averages for ensemble of events:

$$AAE_{\hat{T}} = \frac{1}{N_e}\sum_{i=1}^{N_e}\cos^{-1}(\hat{\mathbf{T}}_{i,pred}\cdot\hat{\mathbf{T}}_{i,g}) \tag{6.24}$$

$$AAE_{\Omega} = \frac{1}{N_e}\sum_{i=1}^{N_e}\cos^{-1}(\mathbf{\Omega}_{i,pred}\cdot\mathbf{\Omega}_{i,g}) \tag{6.25}$$

where N_e is the number of events. Further, We evaluate the angular velocity's Magnitude Error ME at timestep t_i as:

$$ME(t_i) = |\|\mathbf{\Omega_{pred}}\|_2 - \|\mathbf{\Omega_g}\|_2| \tag{6.26}$$

and its average as:

$$AME = \frac{1}{N_e}\sum_{i=1}^{N_e}|\|\mathbf{\Omega_{i,pred}}\|_2 - \|\mathbf{\Omega_{i,g}}\|_2| \tag{6.27}$$

Finally, we normalize the AME with respect to the ground truth angular velocity to obtain the relative AME:

$$AME_{rel} = \frac{1}{N_e}\sum_{i=1}^{N_e}\frac{|\|\mathbf{\Omega_{i,pred}}\|_2 - \|\mathbf{\Omega_{i,g}}\|_2|}{\|\mathbf{\Omega_g}\|_2} \tag{6.28}$$

6.5 Results

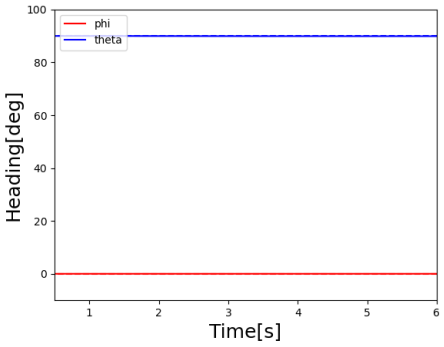
First, we show that our framework can accurately estimate the camera heading for the same planar motion sequences analysed in Section 5.5.3.2. Note that for these sequences

$$\hat{\mathbf{T}}_{\mathbf{g}} = \begin{bmatrix} 0^\circ \\ 90^\circ \end{bmatrix}$$

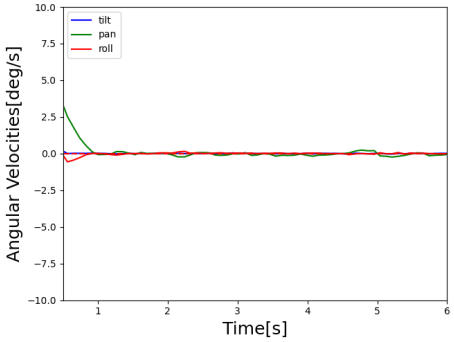
and

$$\mathbf{\Omega_g} = \begin{bmatrix} 0^\circ \\ 0^\circ \\ 0^\circ \end{bmatrix}$$

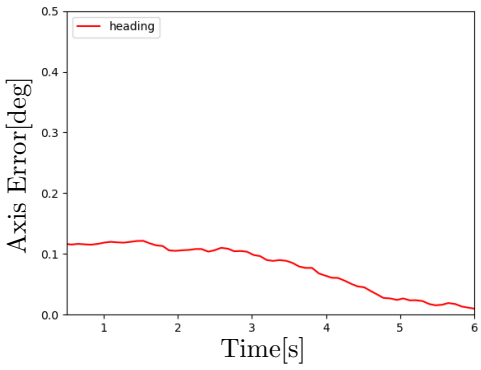
Our predicted heading and angular velocity are shown in solid lines whereas the corresponding ground truth motions are shown in dashed lines in Figs. 6.4(a) and (b) for the *slider far* sequence. The difference between the predicted and ground truth heading are small, Fig. 6.4(a), as quantified in Fig. 5.10(c). The AE approaches zero over time. The predicted angular velocity, Fig. 6.4(b), settles down to the ground truth within 1 second. The results show that our approach can accurately track the ground truth motion on an AES sequence.



(a) Heading



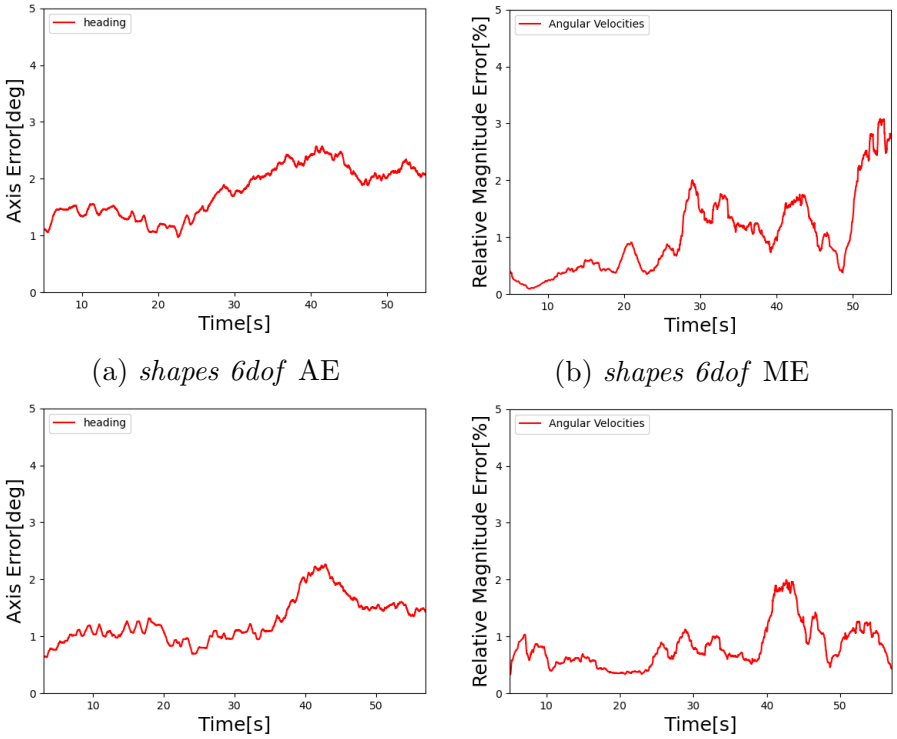
(b) Angular Velocities



(c) Heading error

Figure 6.4: The predicted (solid lines) and ground truth (dashed lines) (a) heading and (b) angular velocity for the *slider far* sequence and (c) the corresponding heading AE.

Next, we test our approach's ability to accurately estimate motion in general motion sequences. We show in Fig. 6.5 the AE and relative ME evaluated over time for *shapes 6dof* and *hdr poster* sequences. The results show, Figs. 6.5(a) and (b), that our approach can accurately track the camera heading with an $AAE < 2\,\text{deg}$ and angular velocity with an $AME_{rel} < 1\%$ for *shapes 6dof* sequence. It also shows that our approach performance on par for the high dynamic range sequence *hdr poster*, Figs. 6.5(c) and (d), with an $AAE < 1.5\,\text{deg}$ and an $AME_{rel} < 1\%$. Similar results were obtained for the other four general motion sequences.



(a) *shapes 6dof* AE (b) *shapes 6dof* ME

(c) *hdr poster* AE (d) *hdr poster* ME

Figure 6.5: Samples of the heading axis error, and relative angular velocities error over time for the *shapes 6dof* and *hdr poster* sequences.

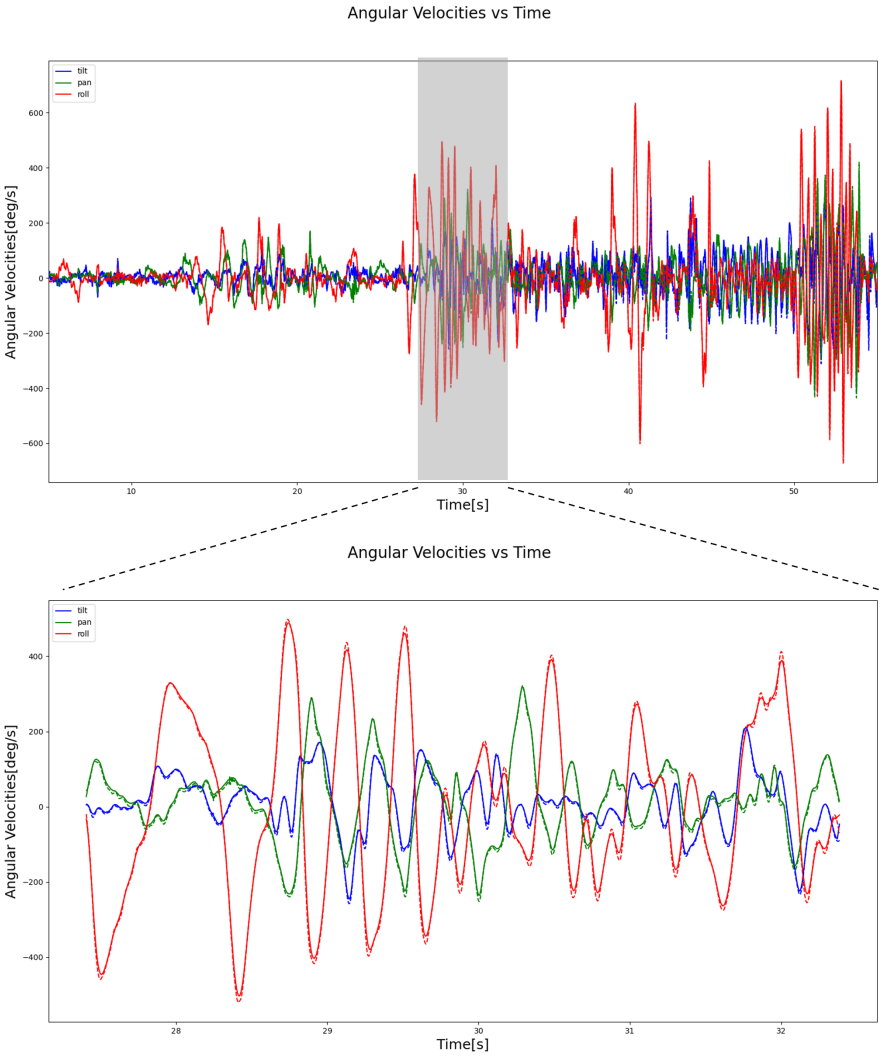


Figure 6.6: Comparison of tilt (blue), pan (green), and roll (red) ground truth angular velocities (dashed lines) to our predictions (solid lines) for the *shapes 6dof* sequence. The top panel shows the full sequence and the bottom panel zooms-in on the shaded area.

To analyse these results in further detail, we show in Fig. 6.6 the components of the predicted (solid lines) and ground truth (dashed lines) angular velocity vectors for the *shapes 6dof* sequence. The top panel of the figure shows their time evolution over the entire sequence. The bottom panel of the figure zooms-in over a 7-seconds segment shaded in the top panel. The predicted angular velocities follow closely the ground truth expect at large extremities. This is true even for very large angular velocities, at the order of hundreds of deg/s. This shows that event cameras equipped with our motion estimation technique can be used as an event-based gyroscope.

We evaluated the AAE for the translational heading and the AAE, AME and relative AME for the angular velocity estimated using our approach over time for the four planar sequences described in Section 5.5.1.2 and the 6 general motion sequences described above. The results are listed in Table 6.1. For the planar motion sequences our approach is able of accurately tracking the camera heading with AAE below 0.3 deg for all four sequences. This shows that the forward prediction objective function can accurately estimate the heading of planar motions but not their magnitudes.

Furthermore, the translational and angular velocity headings AAE for all general motion sequences, expect for the textured *boxes rotation*, are less than 2 deg. Similarly, the AME_{rel} is less than 1% for all general motion sequences except for the textured *boxes rotation*. We note that while the absolute error in the angular velocity magnitude AME was on the order of a few deg/s, for instance 4.65 deg/s on the *shapes 6dof* sequence, the underlying ground truth velocity in those instances saw large excursions, reaching a maximum of 700 deg/s for this sequence (see Fig. 6.6) and , therefore, those errors are in fact relatively small.

We further evaluated the percentage of outlier events for all general motion sequences. Since the ground truth data is sampled at a much lower rate than the AES (200 Hz), we used linear interpolation to create a continuous estimate of the ground truth between each pair of readings. An event was

Table 6.1: The average axis error AAE for translational and angular velocity, the average magnitude error AME and relative average magnitude error AME_{rel} for angular velocity, and the % of outlier events for 10 sequences of the Event-Camera Dataset and Simulator [120].

Sequence	Translational Heading	Angular Velocities			Candidates
	AAE[deg]	AAE[deg]	AME[deg/s]	AME_{rel}[%]	Outliers[%]
slider far	0.07	0.11	N/A	N/A	N/A
slider close	0.072	0.16	N/A	N/A	N/A
slider hdr far	0.24	0.1	N/A	N/A	N/A
slider hdr close	0.23	0.11	N/A	N/A	N/A
shapes rotation	0.39	1.12	3.23	0.73	4.55
shapes translation	0.9	0.79	1.63	0.36	3.38
shapes 6dof	1.7	1.84	4.65	0.94	5.82
poster 6dof	1.55	1.81	4.22	0.91	5.79
hdr poster	1.21	1.43	3.76	0.84	4.9
boxes rotation	5.7	5.36	11.9	3.32	10.6

considered an outlier, if its $ME \geq 5\%$ with respect to the continuous estimate of the ground truth. The percentage of outlier events are listed in the last column of Table 6.1 for all general motion sequences. The results show that our forward prediction objective function associates more then 90% of the events correctly while rejecting most of the outlier events that exacerbate errors. We attribute the superior performance of our approach to its ability to correctly associate events and their priors along event tracks.

6.6 Discussion

6.6.1 Weighting of the Time Penalty

We revisit the feasibility of tuning the time penalty to eliminate ambiguities in the forward prediction objective function. We evaluated the AAE, AME_{rel}, and the percentage of outliers for a range of values of r_t stretching

from zero to 30,000. The results for 5 selected values of r_t on the *shapes 6dof* sequence are shown in Table 6.2.

Table 6.2: The impact of the time penalty weight for motion estimation on the *shapes 6dof* sequence.

Time Penalty	Translational Heading	Angular Velocities	Candidates
$\times 10^4 [pixels/s]$	AAE[deg]	AME_{rel}[%]	Outliers[%]
0	3.81	3.37	11
1	1.76	1.06	5.98
1.5	1.7	0.94	5.82
2	2.25	1.3	6.59
2.5	2.49	1.66	7.2

We found that setting r_t to optimal (intermediate) values, between $r_t = 10000$ to $r_t = 15000$, results in an optimal motion accuracy due to a low number of outliers which was below 6%. Abating the time penalty ($r_t = 0$) or setting its weight to small values, compared to optimal weight, decreases of accuracy by 3 folds due to the elevated number of uncorrelated outlier events. Further, setting r_t to larger than optimal values, progressively decreases the accuracy of motion estimation as it results in progressively larger numbers of outliers. This is due to the time penalty weight overriding the bilinear constraint term. Similar results were obtained on all the other sequences. Therefore, the time penalty weight should be tuned to optimal values. Ablating ($r_t = 0$), undertuning or overtuning it is undesirable.

6.6.2 Comparison to the State-of-the-Art

In the absence of open-source implementations of event-based general motion estimation techniques and given the unavailability of the sequences used to test techniques in the literature, we compare the global performance metrics of our approach to the corresponding metrics of those techniques. We start

by comparing to the pose estimation technique of Reverter et al. [147] who report relative errors in the camera pose. Assuming that the differentiation of the camera pose to obtain velocity will incur Gaussian noise, we expect their estimate of angular velocity to have the same AME_{rel} as that of the pose itself. They report an AME_{rel} in the range of 1.3–2% for 4 sequences of simultaneously moving objects and an AME_{rel} of 4% for a rapid motion sequence. Their sequences were not textured and our comparable performance for this case is at less than 1%.

Next, we compare to the contrast maximization method of Peng et al. [138] and the SLAM-based approaches of Wang et al. [168]. Since the magnitude of angular velocity in these sequences vary, we follow Liu et al. [99] in establishing a metric that can compare performance among these sequences defined as follows:

$$AME_N = \frac{1}{N_e} \sum_{i=1}^{N_e} \frac{|\|\boldsymbol{\Omega}_{\mathbf{i,pred}}\|_2 - \|\boldsymbol{\Omega}_{\mathbf{i,g}}\|_2|}{\|\boldsymbol{\Omega}_{\mathbf{max,g}}\|_2} \tag{6.29}$$

where $\boldsymbol{\Omega}_{\mathbf{max,g}}$ is the maximum ground truth angular velocity in the sequence. Since the aforementioned sequences were not textured we compared them to the first five untextured general motion sequences in our case. The AME_N for the contrast maximisation approach [138] was in the range of 1.3–1.4%, and for the SLAM-based approach [168] was 2.1–3.7%. Our approach outperforms them with an AME_N of 0.5–0.75% over those five sequences.

Finally, we compare our approach to two recently developed approaches limited to estimation of 3D rotation, namely CM [62] and its extension CMBnB [99]. Since CM and CMBnB were evaluated on the textured *boxes rotation* sequence, we compare our results for that sequence to theirs in Table 6.3. Our approach show a notably better accuracy with an AME_N of 1.78% compared to 3.2% and 2.68% for CM [62] and CMBnB [99], respectively.

We conclude that our approach outperforms the state-of-the-art for both textured and untextured sequences. This shows that consensus voting of

Table 6.3: Comparison of the AME and AME_N of our approach to CM and CMBnB for the *boxes rotation* sequence [120].

Method	AME[deg/s]	AME_N[%]
This Approach	**11.9**	**1.78**
CM [62]	21.41	3.2
CMBnB [99]	17.97	2.68

event tracks via asynchronous processing is a more effective motion estimation technique than those that accumulate events into image structures. Our approach avoids the spatial smoothness assumption which is violated at object boundaries and discontinuities specially at low speeds where events can be very sparse, thereby deteriorating the accuracy of their motion estimation. In fact, our approach is especially suited for sparse scenes where the accumulation approaches suffer a lower information rate forcing them to either estimate motion from a sparse scene resulting in inferior accuracy or accumulate events over longer windows thereby excluding faster motions. At higher speeds, our approach has proven more effective at solving the data association problem. Finally, the pure 3D rotational approaches ignore translation which results in a lower motion accuracy, unlike the case for our approach.

6.6.3 Rapid Motion Limitations

In this section, we test the limits at which the velocity constancy assumption starts to undermine the accuracy of our egomotion estimation. Towards that end, we evaluate in Fig. 6.7 the relative ME for each of the angular velocity components of the *shapes 6dof* sequence shown in the top panel of Fig. 6.6. The relative error is limited to less than 1% for all rotations with a magnitude less then 200 deg/s. Error grows as our algorithm encounters large motion excursions but it settles down below 1% as they end. This behaviour is consistent for all three angular velocity components. Errors

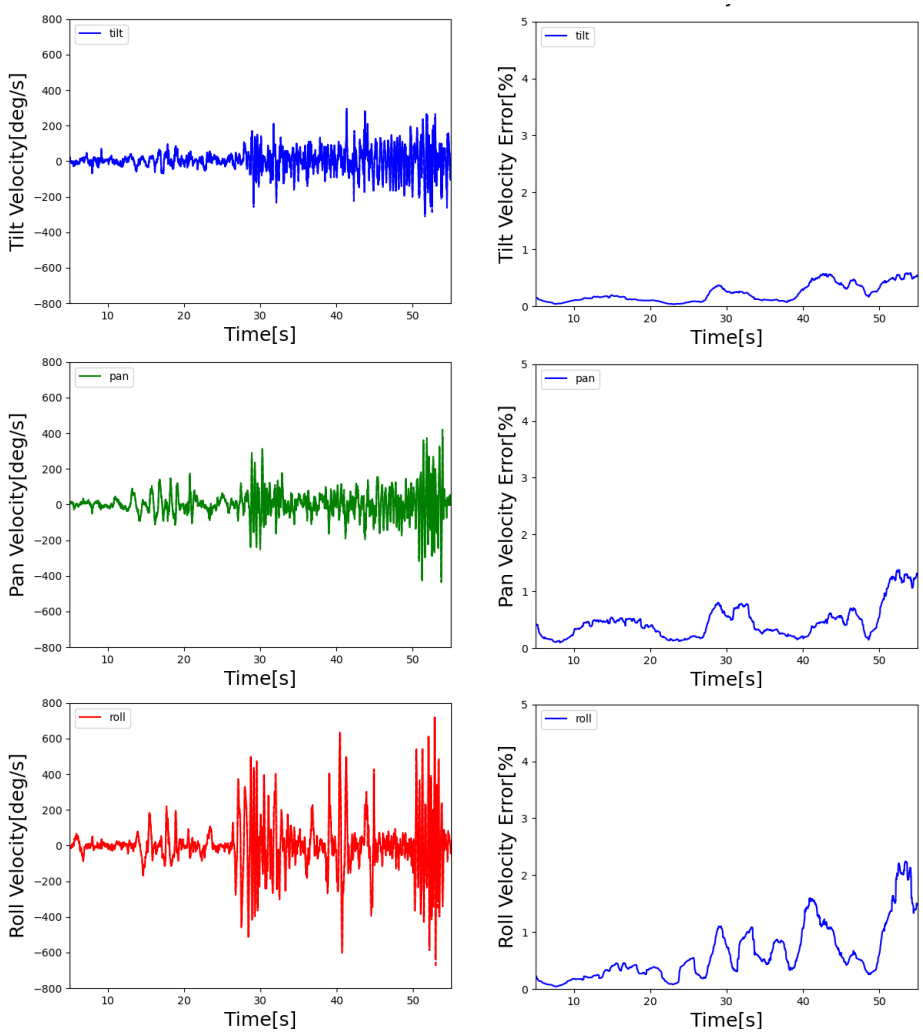


Figure 6.7: The tilt (blue), pan (green), and roll (red) rotations of the *shapes 6dof* sequence and the corresponding relative ME as functions of time.

grow proportionally to the magnitude of those excursions as can been seen by comparing the lager excursions of roll to the relatively smaller ones of pan and tilt. Similar results were obtained for all other five general motion sequences.

We also note that the AME_N for our approach was less than that of the comparable general motion approaches [138, 168] eventhough our sequences included larger angular velocities excursions reaching maxima varying from 620 to 940 deg/s, whereas the maximum angular velocity in the sequences handled by those approaches did not exceed 45 deg/s. Combined, these results indicate that our approach maintains accuracy at much higher angular speeds.

6.6.4 Texture Limitations

Textured scenes produce denser event streams (more events per second) undermining the efficiency of event cameras. At the limit where all the pixels in an event camera fire at the same time, it becomes equivalent to a traditional camera with a similar latency due to hardware limitations. In these cases, the arbiter (see Appendix A) decreases the bandwidth of the camera, decides the order of events transmission, and assigns them incorrect timestamps. As a result, correct candidate events may be delayed to lie outside our temporal window $m\delta t$, and missed or the data association process fails due to events carrying the wrong timestamp. The impact of texture on performance can be observed in the *boxes rotation* sequence which returned higher errors than all the other sequences, Table 6.1. The event rate for this sequence was four times that of the other general motion sequences.

Figure 6.8 shows the relative ME for the *boxes rotation* sequence. The presence of texture and rapid motion degrades accuracy over the second half of the sequence. However, comparing the ME_{rel} for this sequence to that of the *shapes 6dof*, Fig. 6.7, shows that while a similar level of rapid motions in the latter sequence generated $ME_{re]} < 3\%$, in this sequence it generated $ME_{rel} > 6\%$. The texture is the factor exacerbating those errors. Further, we note that the percentage of outlier events for this sequence was double that rate for all sparse general motion sequences. The elevated rate of the data association failure is due to the arbiter interference discussed above.

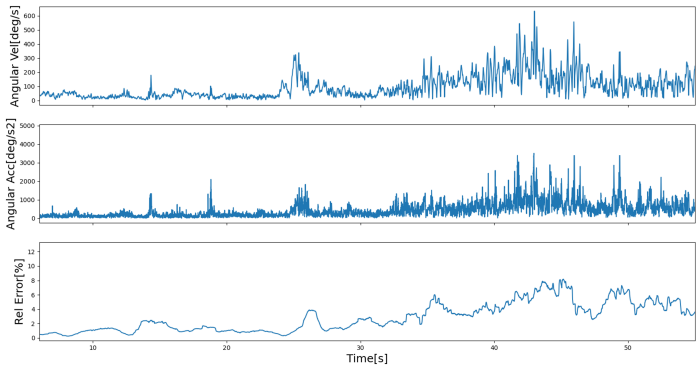


Figure 6.8: Ground truth velocity, ground truth acceleration, and the relative ME for the *boxes rotation* sequence.

However, eventhough the texture sequence had an increased level of error, our approach was still able to estimate the motion accurately with an AME_{rel} of 3.32%.

6.6.5 Edge Normal to the Flow

The forward prediction objective function encountered an anomaly in scenes that include edges or other patterns perpendicular to the direction of motion. An example of this situation is the *slider far* sequence captured while the camera moves parallel to the scene shown in Fig 6.9. In these cases, the objective function can develop two local minima as events fire with very close timestamps. As a result, the objective function encounters an ambiguity whether the direction of motion is along the actual path or the sequence of closely spaced events up the edge. To minimize the impact of this ambiguity, our algorithm implements a two-step investigation. First, it investigates whether the timestamps of the candidate events lie within a very small time window. Second, for those events that lie within that time window, it finds their time priors and again investigates whether those priors lie within a

similar-sized time window. If both conditions are met, it concludes that those events belong to a vertical pattern and the motion they suggest is spurious and, therefore, they are removed from the candidate event pool. We found that while this implementation was able to limit the impact of edges on the accuracy of motion estimation. For instance, the AAE increased from 0.07 deg for the other planar motion sequence to 2.18 for the *slider far* sequence.

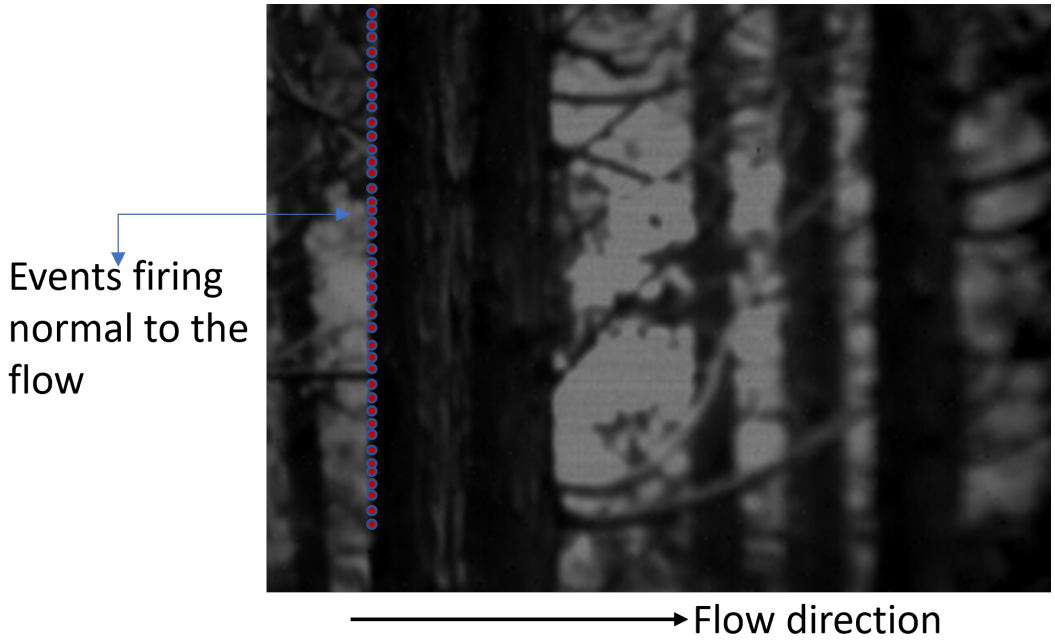


Figure 6.9: Events firing along an edge normal to the flow direction.

6.6.6 Processing Time

All experiments were ran on a desktop PC consisting of one GPU (NVIDIA GeForce RTX 2070 Super) and one CPU (Intel Core i7 10700k 5.1GHz 8-core). We used a brute force implementation of the particle filter which processes each event individually. We found that the general motion frame-

work was more efficient then the planar motion framework since all candidate events were evaluated only once against all particles. For the former framework, events processing times were on the order of a few μs when using a particle filter consisting of 100 particles. Processing times increased linearly with the number of particles reaching tens of μs for 1500 particles. We estimate that for 100-particle filter our framework as implemented can process more than 100 k-events/s in real-time. Using more particles will degrade real-time capabilities to tens of k-events/s.

6.7 Conclusion

In this chapter, we validated the thesis hypothesis by presenting a new voting based method which relies only on the consistency of camera motion and the correct data association of events and their previous firing along corresponding tracks. We showed that the proposed approach works even in the case of arbitrary camera motion and arbitrary scene depth in which a single data association can vote for multiple motions. Our results show that our approach significantly outperforms other approaches in terms of accuracy and robustness and can achieve high accuracy in situations of rapid motion and high dynamic range in the presence of unknown depth. Finally, we characterized situations that limit its accuracy, mainly in highly textured environments, very fast motion and events firing along edge normal to the flow.

Chapter 7

Conclusions and Future Work

Motion estimation is an essential problem at the core of various applications; from robotics to self driving cars to multimedia use such as augmented reality and gaming, the accuracy and robustness of motion estimation relies on the cheap and low cost motion sensors that are commonly used. Event cameras are new emerging motion sensors within this category with characteristics which make them offer better alternatives to motion estimation that are worth exploring. While viable event-based motion estimation solutions have been emerging, there is still ample room for improvements.

In this thesis, we wanted to solve the event-based motion estimation problem in a fundamentally different way, by hypothesising that in the case of a dominant motion between the camera and the scene, consistency with the dominant motion is sufficient for correct data association of events and their previous firings along event tracks and would result in more accurate and robust motion estimation.

Towards that, we presented two novel voting based methods that rely on considering all potential data association candidates that are consistent with a single camera motion for candidates evaluation by handling each event individually without assuming any relationship to its neighbors beyond the common camera motion.

The first method projects an event backward then search over all candidate events around its projection to select the one that is consistent with the camera motion. We validated the posited hypothesis first mathematically by conducting a mathematical analysis, then experimentally by exploiting the proposed method in a particle filter framework for the simple case of planar motion which yielded motion estimates that were an order of magnitude more accurate than optical flow based approaches.

The consensus based method was extended to solve the general case of event-based motion estimation where the camera undergoes an arbitrary motion in the presence of unknown scene depth. This was achieved by presenting a novel voting method which projects forward a set of candidate image velocities from the data association, then perform a search over all those candidates to vote for the one that is most consistent with the camera motion. We validated the posited hypothesis experimentally in a particle filter based motion estimation system on a challenging AES dataset, where we showed that our approach can significantly outperform the state-of-the-art in terms of accuracy and robustness.

Therefore, we proved that consistency with a single camera motion can be used to solve the data association problem even in the case of unknown depth. Based on that, we provided approaches that led to superior planar and general motion estimation. Putting all the work of this thesis together, we presented a system contribution towards a unified asynchronous event-based motion estimation. Our approach benefits from the event cameras advantages where it should be used in situations of rapid motion, high dynamic range, strong illumination changes, and in sparse scenes. However, it is not recommended to use it in situations that limits its accuracy, namely in textured environments where events fire at a high rate which decreases the bandwidth of the camera and makes realizing real-time processing harder, and in scenes with repetitive patterns such as events firing normal to the flow direction.

Moving forward, there are a couple of interesting aspects to improve. The next step would tackle improving our particle filter. We used a particle filter as the motion estimation framework. However, the brute force nature of a particle filter requires monotonically increasing processing time with respect to the number of particles being used. Our current particle filter is brute force and need improvements to ensure it works in real-time applications. One possible improvement to realize real-time processing is to directly use the particles from the forward predictions, each prediction becomes a particle if no particle is close enough to it, or it uses a voting mechanism to vote to its closest (valid) particles. So particles are created from forward predictions. The voting mechanism is based on their weight score, some of them are bad and should eventually die, and some are good (valid) and should get updated.

Another improvement would be to use deep learning as the motion estimation framework. In particular, we will need a network that can handle each event individually preserving the asynchronous nature of our approaches while building a history of events, such as Recurrent Neural Networks (RNNs) [149, 64]. However, the challenge is in building such an architecture. One possible solution would require an embedding layer, a set of deconvolutional layers, and a LSTM component to learn the temporal aspect (e.g., PhasedLSTM [126]) that use our objectives functions as unsupervised loss functions.

Finally, a by product of using the forward prediction is that we can determine not only the camera motion but the optical flow corresponding to each event. After the estimate of the camera's egomotion is obtained, the flow prediction that is the most consistent with it can be selected as the estimate of the optical flow. This optical flow could be use to analyze the assumption of local smoothness and how often it is broken in the scenario of a camera moving in a natural scene. Additionally, an optical flow computed in a similar way but using a ground truth camera motion instead of estimate motion could be used to analyze the noise statistics and dynamics in the AES.

APPENDICES

Appendix A

Hardware Limitation

Our findings from working on a project involving tracking the vibrations of a micro-structural dynamics component, confirms that textured objects or environments, will be prone to major hardware limitation. Rich texture will trigger a very big number of events (could be all pixels) to fire at the same time. When all pixels fire at once we have a problem of timestamps loss, as the pixels will not be sent to the FPGA in order. The latter is due to the AES bandwidth: for instance, if 128×128 pixels are firing at the same time considering the 1 million events per second (see specs in Fig. 2.1), therefore the maximum bandwidth becomes 60Hz or $pixels/s$. The latter implies that the AE camera is now as slow as regular cameras. The problem is caused by a hardware chip called the arbiter shown in Fig. A.1, and summarized as follows [20]:

1 If all pixels are fire at once: only one pixel is allowed to pass at a time. First the pixels send requests to the row arbiter, which randomly decides the row that will be allowed to go first.

2 Then all pixels that fired on this row send requests to the column arbiter.

3 The column arbiter would go through them one by one.

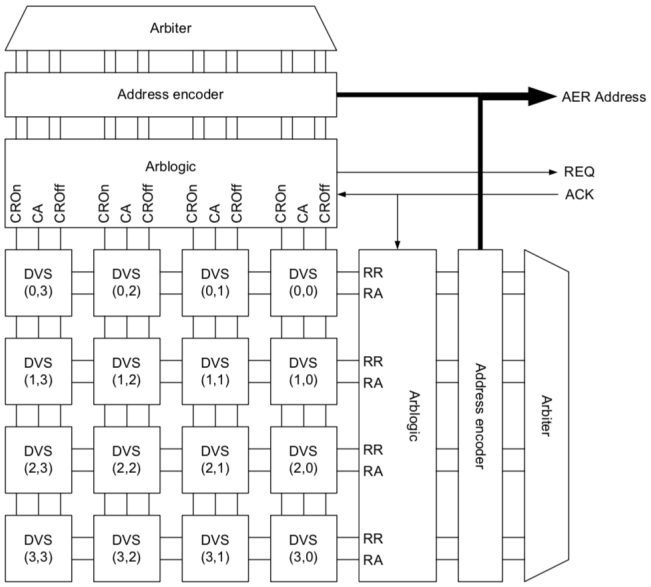


Figure A.1: DVS block diagram circuitry. Figure taken from [20].

4 When all the pixels on the selected row are sent, the row arbiter would select another row.

Thus, when many pixels fire at once, they are always processed row by row. The arbiter is the cause of the 1Meps which causes the delay of $1/60s$ if all the pixels fire at once.

Appendix B

Event-based Data Generator

This is synthetic dataset to study the motion estimation and its limitations.

B.1 OF Ground Truth Generation

Simulation goal is to provide at each instant t_i, the optical flow of a NxN grid assuming the camera is facing an infinite wall of depth Z.

T_i: Translation at time instant t_i. T_i would be made to be constant, linear or, follows a sinusoidal pattern of the form:

$$T_i = f(t_i) = (asin(r_c t_i), bsin(r_c t_i - \theta)) \tag{B.1}$$

so it varies smoothly within $-a$ and a for T_i^x and $-b$ and b for T_i^y. r_c controls the sine movement a b and r_c can be chosen as the maximum desired velocity

Under a general simulation assumption the data can be generated as follow:

- At each instant T_i:

1. Generate for each grid point (x_k, y_k) a 3D point (X_k, Y_k, Z_k):

$$X_k = (x_k - c_x)\frac{Z_k}{f} \tag{B.2}$$

$$Y_k = (y_k - c_y)\frac{Z_k}{f} \tag{B.3}$$

   c_x and c_y can be considered $= N/2$

2. Translate the point by $T_i = f(i)$:

$$X_T = X_k + T_i^x \tag{B.4}$$

$$Y_T = Y_k + T_i^y \tag{B.5}$$

$$Z_T = Z_k \tag{B.6}$$

3. Project back to get (x_T, y_T):

$$x_T = fX_T/Z + c_x = x_k + f\frac{T_i x}{Z} \tag{B.7}$$

$$y_T = fY_T/Z + c_y = y_k + f\frac{T_i y}{Z} \tag{B.8}$$

4. Determine optical flow as the difference between (x_T, y_T) and (x_k, y_k)

- Note that the x_t and y_t could be obtained as expected without the projection for 3D, however the simulation should be made so it generalizes to different cases of transformations. So for a different transformation what it needs is the expression of X_T and Y_T as functions of X_k and Y_k: $[X_T, Y_T, Z_T] = F_i(X_k, Y_k, Z_k) = R_i * [X_k; Y_k; Z_k] + T_i$, Where R_i is the rotation matrix corresponding to the rotational velocity ω_i

B.1.0.1 Important Parameters

   ⋆ $Z = Z_K$, f, a, b and r_c, are chosen in such a way to make the optical flow in the order of few pixels

- ⋆ Grids will be 32×32 to capture the OF
- ⋆ timesteps increment will be linear of $\Delta t = 50\mu s$: at each timestep t_i in Eq. B.1, $t_i = t_{i-1} + \Delta t$
- ⋆ the instantaneous OF (u,v) at each timestep has the generalized units of pixels/frame i.e. pixels/$50\mu s$. The average velocity will be over the whole sequence to make the make sure we have few pixels movements per sequence.

B.2 Events Generation

Once we have the optical flow at each grid, we have to generate events from them. To obtain the events we need to have intensity values. Towards that, we use a large dense image as shown in Fig B.1. We take a small patch grid (reference grid) $32x32$ at t_0 which results in an intensity grid I_0.

Figure B.1: 30000x17000 Large Dense Image

Then we warp I_0 with the ground truth optical flow values to predict the intensities at time $t_0 + 1$ and generate I_1. This is exactly like moving the camera parallel to the main image (our static scene here) with a know

velocity (ground truth) starting from the reference grid at t_0. Points that do not have predicted intensities in I_1 get assigned new random intensities. The process gets repeated for each time step t to obtain I_t for all the considered sequences.

B.2.0.1 Image Warping

To do the image warping from I_0 to I_1, we start from every pixel of I_1 and determine its corresponding ancestor at 0:

$$[\hat{x}_0, \hat{y}_0] = [x_1, y_1] - [u, v] \tag{B.9}$$

Then the intensity of the pixel $I_1(x_1, y_1)$ is bilinearly interpolated from the pixels neighbours of $[\hat{x}_0, \hat{y}_0]$

let

$$\begin{aligned} i_m &= floor(\hat{x}_0) \\ j_m &= floor(\hat{y}_0) \\ i_p &= ceil(\hat{x}_0) \\ j_p &= ceil(\hat{y_0}) \end{aligned}$$

and

$$\begin{aligned} d_0 &= i_p - \hat{x}_0 \\ d_1 &= \hat{x}_0 - i_m \\ d_2 &= j_p - \hat{y}_0 \\ d_3 &= \hat{y}_0] - j_m \end{aligned}$$

if none of $[i_m, j_m]$, $[i_m, j_p], [i_p, j_m]$, $[i_p, j_p]$ are within the bounds of I_0 then $I_1([x_1, y_1])$ can be considered as a new pixel and assigned a new value.

Alternatively, if $[\hat{x}_0, \hat{y}_0]$ itself is outside the bounds of I_0 then $I_1([x_1, y_1])$ can be considered as a new pixel and assigned a new value.

Let

$$\begin{aligned} v_0 &= I_0([i_m, j_m]) \\ v_1 &= I_0([i_p, j_m]) \\ v_2 &= I_0([i_m, j_p]) \\ v_3 &= I_0(i_p, j_p]) \end{aligned}$$

Then, the interpolated value would be:

$$v = d_2 \times (v_0 \times d_1 + v_1 \times d_0) + d_3 \times (v_2 \times d_1 + v_3 \times d_0) \tag{B.10}$$

Let δI_t denote the difference $I_t - I_{t-1}$ then for each pixel x, y for which the absolute value of δI_t is greater than a certain threshold ϵ a corresponding event is generated with the following value:

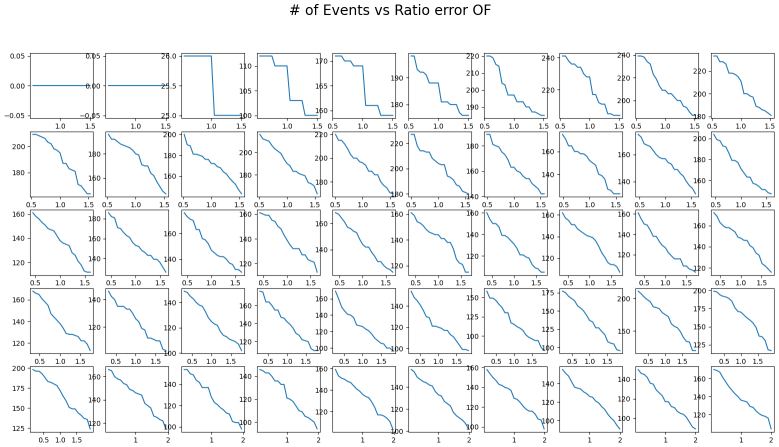
$$[x, y, t, sgn(\delta I_t(x, y))], \tag{B.11}$$

where sgn is the sign function.

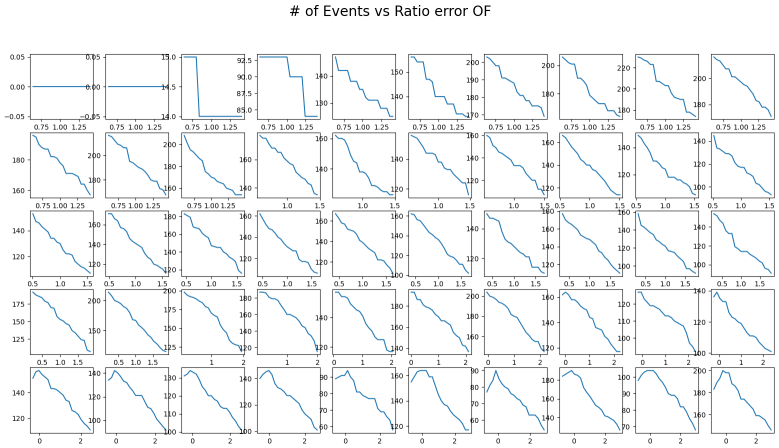
Appendix C

Initialization

We show that the distance-based method in Eq. 5.3 needs to gather events history to initialize properly when processing each event individually. We consider a camera moving with two different sinusoidal motion of 1 and 2 peak to peak velocities respectively, and illustrate the number of events against the ratio of error in motion (1 being the correct motion). Fig. C.1 shows that the first $3-4$ frames (timesteps) do not have enough history of events with $\leq 20\%$ average number of events, resulting in the inability to properly minimize the objective function.



(a) 1 peak to peak



(b) 2 peak to peak

Figure C.1: Events against the ratio error in OF for a sample of 50 frames for a camera moving with two different sinusoidal motion of 1 and 2 peak to peak velocities respectively.

References

[1] Himanshu Akolkar, Sio Hoi Ieng, and Ryad Benosman. Real-time high speed motion prediction using fast aperture-robust event-driven visual flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[2] Himanshu Akolkar, SioHoi Ieng, and Ryad Benosman. See before you see: Real-time high speed motion prediction using fast aperture-robust event-driven visual flow. *arXiv preprint arXiv:1811.11135*, 2018.

[3] Mohammed Almatrafi, Raymond Baldwin, Kiyoharu Aizawa, and Keigo Hirakawa. Distance surface for event-based optical flow. *IEEE transactions on pattern analysis and machine intelligence*, 42(7):1547–1556, 2020.

[4] Mohammed Almatrafi and Keigo Hirakawa. Davis camera optical flow. *IEEE Transactions on Computational Imaging*, 6:396–407, 2019.

[5] Ignacio Alzugaray and Margarita Chli. Asynchronous multi-hypothesis tracking of features with event cameras. In *2019 International Conference on 3D Vision (3DV)*, pages 269–278. IEEE, 2019.

[6] Ignacio Alzugaray Lopez and Margarita Chli. Haste: multi-hypothesis asynchronous speeded-up tracking of events. In *31st British Machine Vision Virtual Conference (BMVC 2020)*, page 744. ETH Zurich, Institute of Robotics and Intelligent Systems, 2020.

[7] Padmanabhan Anandan. Measuring visual motion from image sequences. 1987.

[8] Padmanabhan Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, 1989.

[9] Alex M Andrew. Multiple view geometry in computer vision. *Kybernetes*, 2001.

[10] Jithendar Anumula, Daniel Neil, Tobi Delbruck, and Shih-Chii Liu. Feature representations for neuromorphic audio spike streams. *Frontiers in neuroscience*, 12:23, 2018.

[11] Christian Bailer, Bertram Taetz, and Didier Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 4015–4023, 2015.

[12] 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.

[13] Connelly Barnes, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. The generalized patchmatch correspondence algorithm. In *European Conference on Computer Vision*, pages 29–43. Springer, 2010.

[14] Francisco Barranco, Cornelia Fermüller, and Yiannis Aloimonos. Contour motion estimation for asynchronous event-driven cameras. *Proceedings of the IEEE*, 102(10):1537–1556, 2014.

[15] Francisco Barranco, Cornelia Fermuller, and Yiannis Aloimonos. Bio-inspired motion estimation with event-driven sensors. In *Interna-*

tional Work-Conference on Artificial Neural Networks, pages 309–321. Springer, 2015.

[16] John L Barron, David J Fleet, and Steven S Beauchemin. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77, 1994.

[17] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

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

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

[20] Raphael Berner. *Building-blocks for event-based vision sensors*. PhD thesis, ETH Zurich, 2011.

[21] Michael J Black and Paul Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer vision and image understanding*, 63(1):75–104, 1996.

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

[23] Tobias Brosch, Stephan Tschechne, and Heiko Neumann. On event-based optical flow detection. *Frontiers in neuroscience*, 9:137, 2015.

[24] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, pages 25–36. Springer, 2004.

[25] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):500–513, 2011.

[26] Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International journal of computer vision*, 61(3):211–231, 2005.

[27] Anna R Bruss and Berthold KP Horn. Passive navigation. *Computer Vision, Graphics, and Image Processing*, 21(1):3–20, 1983.

[28] Samuel Bryner, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. Event-based, direct camera tracking from a photometric 3d map using nonlinear optimization. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 325–331. IEEE, 2019.

[29] E Caetano, S Silva, and J Bateira. A vision system for vibration monitoring of civil engineering structures. *Experimental Techniques*, 35(4):74–82, 2011.

[30] O Cakar and KY Sanliturk. Elimination of transducer mass loading effects from frequency response functions. *Mechanical Systems and Signal Processing*, 19(1):87–104, 2005.

[31] Marco Cannici, Marco Ciccone, Andrea Romanoni, and Matteo Matteucci. Event-based convolutional networks for object detection in neuromorphic cameras. *arXiv preprint arXiv:1805.07931*, 2018.

[32] João Carneiro, Sio-Hoi Ieng, Christoph Posch, and Ryad Benosman. Event-based 3d reconstruction from neuromorphic retinas. *Neural Networks*, 45:27–38, 2013.

[33] P Castellini, M Martarelli, and EP Tomasini. Laser doppler vibrometry: Development of advanced solutions answering to technology's needs. *Mechanical Systems and Signal Processing*, 20(6):1265–1285, 2006.

[34] 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.

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

[36] Andrea Censi, Jonas Strubel, Christian Brandli, Tobi Delbruck, and Davide Scaramuzza. Low-latency localization by active led markers tracking using a dynamic vision sensor. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 891–898. IEEE, 2013.

[37] William Chamorro, Juan Andrade-Cetto, and Joan Solà. High speed event camera tracking. *arXiv preprint arXiv:2010.02771*, 2020.

[38] Justin G Chen, Neal Wadhwa, Young-Jin Cha, Frédo Durand, William T Freeman, and Oral Buyukozturk. Modal identification of simple structures with high-speed video using motion magnification. *Journal of Sound and Vibration*, 345:58–71, 2015.

[39] Nicholas FY Chen. Pseudo-labels for supervised learning on dynamic vision sensor data, applied to object detection under ego-motion. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 644–653, 2018.

[40] Xavier Clady, Charles Clercq, Sio-Hoi Ieng, Fouzhan Houseini, Marco Randazzo, Lorenzo Natale, Chiara Bartolozzi, and Ryad Benosman. Asynchronous visual event-based time-to-contact. *Neuromorphic Engineering Systems and Applications*, 51, 2015.

[41] 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.

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

[43] Matthew Cook, Luca Gugelmann, Florian Jug, Christoph Krautz, and Angelika Steger. Interacting maps for fast visual interpretation. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 770–776. IEEE, 2011.

[44] 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.

[45] Tobi Delbruck and Manuel Lang. Robotic goalie with 3 ms reaction time at 4% cpu load using event-based dynamic vision sensor. *Neuromorphic Engineering Systems and Applications*, page 16, 2015.

[46] Tobi Delbruck and Patrick Lichtsteiner. Fast sensory motor control based on event-based hybrid neuromorphic-procedural system. In *2007*

IEEE international symposium on circuits and systems, pages 845–848. IEEE, 2007.

[47] Tobi Delbruck, Vicente Villanueva, and Luca Longinotti. Integration of dynamic vision sensor with inertial measurement unit for electronically stabilized event-based vision. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2636–2639. IEEE, 2014.

[48] D Di Maio and DJ Ewins. Continuous scan, a method for performing modal testing using meaningful measurement parameters; part i. *Mechanical Systems and Signal Processing*, 25(8):3027–3042, 2011.

[49] Scott W Doebling, Charles R Farrar, Michael B Prime, et al. A summary review of vibration-based damage identification methods. *Shock and vibration digest*, 30(2):91–105, 1998.

[50] Charles Dorn, Sudeep Dasari, Yongchao Yang, Garrett Kenyon, Paul Welch, and David Mascareñas. Efficient full-field operational modal analysis using neuromorphic event-based imaging. In *Shock & Vibration, Aircraft/Aerospace, Energy Harvesting, Acoustics & Optics, Volume 9*, pages 97–103. Springer, 2017.

[51] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.

[52] Randal Douc and Olivier Cappé. Comparison of resampling schemes for particle filtering. In *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69. IEEE, 2005.

[53] David Drazen, Patrick Lichtsteiner, Philipp Häfliger, Tobi Delbrück, and Atle Jensen. Toward real-time particle tracking using an event-based dynamic vision sensor. *Experiments in Fluids*, 51(5):1465–1469, 2011.

[54] David Drazen, Patrick Lichtsteiner, Philipp Häfliger, Tobi Delbrück, and Atle Jensen. Toward real-time particle tracking using an event-based dynamic vision sensor. *Experiments in Fluids*, 51(5):1465, 2011.

[55] David J Ewins. *Modal testing: theory and practice*, volume 15. Research studies press Letchworth, 1984.

[56] Wei Fan and Pizhong Qiao. Vibration-based damage identification methods: a review and comparative study. *Structural Health Monitoring*, 10(1):83–111, 2011.

[57] Clément Farabet, Berin Martini, Benoit Corda, Polina Akselrod, Eugenio Culurciello, and Yann LeCun. Neuflow: A runtime reconfigurable dataflow processor for vision. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2011 IEEE Computer Society Conference on*, pages 109–116. IEEE, 2011.

[58] David J Fleet and Allan D Jepson. Computation of component image velocity from local phase information. *International journal of computer vision*, 5(1):77–104, 1990.

[59] 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.

[60] Guillermo Gallego, Mathias Gehrig, and Davide Scaramuzza. Focus is all you need: Loss functions for event-based vision. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12280–12289, 2019.

[61] Guillermo Gallego, Jon EA Lund, Elias Mueggler, Henri Rebecq, Tobi Delbruck, and Davide Scaramuzza. Event-based, 6-dof camera tracking from photometric depth maps. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2402–2412, 2017.

[62] 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.

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

[64] Henri Gavin. The levenberg-marquardt method for nonlinear least squares curve-fitting problems. *Department of Civil and Environmental Engineering, Duke University*, pages 1–15, 2011.

[65] Daniel Gehrig, Antonio Loquercio, Konstantinos G Derpanis, and Davide Scaramuzza. End-to-end learning of representations for asynchronous event-based data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5633–5643, 2019.

[66] Rohan Ghosh, Abhishek Mishra, Garrick Orchard, and Nitish V Thakor. Real-time object recognition and orientation estimation using an event-based camera and cnn. In *Biomedical Circuits and Systems Conference (BioCAS), 2014 IEEE*, pages 544–547. IEEE, 2014.

[67] Arren Glover and Chiara Bartolozzi. Event-driven ball detection and gaze fixation in clutter. In *2016 IEEE/RSJ International Conference*

on Intelligent Robots and Systems (IROS), pages 2203–2208. IEEE, 2016.

[68] Arren Glover and Chiara Bartolozzi. Event-driven ball detection and gaze fixation in clutter. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 2203–2208. IEEE, 2016.

[69] 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.

[70] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[71] Germain Haessig, Andrew Cassidy, Rodrigo Alvarez, Ryad Benosman, and Garrick Orchard. Spiking optical flow for event-based sensors using ibm's truenorth neurosynaptic system. *IEEE transactions on biomedical circuits and systems*, 12(4):860–870, 2018.

[72] Kaiming He and Jian Sun. Computing nearest-neighbor fields via propagation-assisted kd-trees. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 111–118. IEEE, 2012.

[73] David J Heeger. Model for the extraction of image flow. *JOSA A*, 4(8):1455–1471, 1987.

[74] David J Heeger. Optical flow using spatiotemporal filters. *International journal of computer vision*, 1(4):279–302, 1988.

[75] Mark N Helfrick, Christopher Niezrecki, Peter Avitabile, and Timothy Schmidt. 3d digital image correlation methods for full-field vibration

measurement. *Mechanical Systems and Signal Processing*, 25(3):917–927, 2011.

[76] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[77] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[78] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE conference on computer vision and pattern recognition (CVPR)*, volume 2, page 6, 2017.

[79] Eddy Ilg, Tonmoy Saikia, Margret Keuper, and Thomas Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *European Conference on Computer Vision (ECCV)*, 2018.

[80] Idaku Ishii, Yoshihiro Nakabo, and Masatoshi Ishikawa. Target tracking algorithm for 1 ms visual feedback system using massively parallel processing. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 3, pages 2309–2314. IEEE, 1996.

[81] J Yu Jason, Adam W Harley, and Konstantinos G Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *European Conference on Computer Vision*, pages 3–10. Springer, 2016.

[82] Suren Jayasuriya, Orazio Gallo, Jinwei Gu, Timo Aila, and Jan Kautz. Reconstructing intensity images from binary spatial gradient cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–26, 2017.

[83] Kenichi Kanatani. 3-d interpretation of optical flow by renormalization. *International Journal of Computer Vision*, 11(3):267–282, 1993.

[84] Daniel R Kepple, Daewon Lee, Colin Prepsius, Volkan Isler, Il Memming Park, and Daniel D Lee. Jointly learning visual motion and confidence from local patches in event cameras. In *European Conference on Computer Vision*, pages 500–516. Springer, 2020.

[85] Mina A Khoei, Sio-hoi Ieng, and Ryad Benosman. Asynchronous event-based motion processing: From visual events to probabilistic sensory representation. *Neural computation*, 31(6):1114–1138, 2019.

[86] 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, 2014.

[87] Hanme Kim, Stefan Leutenegger, and Andrew J Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *European Conference on Computer Vision*, pages 349–364. Springer, 2016.

[88] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[89] 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.

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

[91] Cedric Le Gentil, Florian Tschopp, Ignacio Alzugaray, Teresa Vidal-Calleja, Roland Siegwart, and Juan Nieto. Idol: A framework for imu-dvs odometry using lines. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5863–5870. IEEE, 2020.

[92] Jong Jae Lee and Masanobu Shinozuka. A vision-based system for remote sensing of bridge displacement. *Ndt & E International*, 39(5):425–431, 2006.

[93] Hongmin Li, Guoqi Li, Xiangyang Ji, and Luping Shi. Deep representation via convolutional neural network for classification of spatiotemporal event streams. *Neurocomputing*, 299:1–9, 2018.

[94] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. A 128 by 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE journal of solid-state circuits*, 43(2):566–576, 2008.

[95] Robert LiKamWa, Zhen Wang, Aaron Carroll, Felix Xiaozhu Lin, and Lin Zhong. Draining our glass: An energy and heat characterization of google glass. In *Proceedings of 5th Asia-Pacific Workshop on Systems*, pages 1–7, 2014.

[96] Martin Litzenberger, Bernhard Kohn, Ahmed Nabil Belbachir, Nikolaus Donath, Gerhard Gritsch, Heinrich Garn, Christoph Posch, and Stephan Schraml. Estimation of vehicle speed based on asynchronous data from a silicon retina optical sensor. In *2006 IEEE intelligent transportation systems conference*, pages 653–658. IEEE, 2006.

[97] Ce Liu, Antonio Torralba, William T Freeman, Frédo Durand, and Edward H Adelson. Motion magnification. *ACM transactions on graphics (TOG)*, 24(3):519–526, 2005.

[98] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2011.

[99] Daqi Liu, Alvaro Parra, and Tat-Jun Chin. Globally optimal contrast maximisation for event-based motion estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6349–6358, 2020.

[100] Daqi Liu, Alvaro Parra, and Tat-Jun Chin. Spatiotemporal registration for event-based visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4937–4946, 2021.

[101] Min Liu and Tobi Delbruck. Block-matching optical flow for dynamic vision sensors: Algorithm and fpga implementation. In *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*, pages 1–4. IEEE, 2017.

[102] Shih-Chii Liu, Tobi Delbruck, Giacomo Indiveri, Adrian Whatley, and Rodney Douglas. *Event-based neuromorphic systems*. John Wiley & Sons, 2014.

[103] Hugh Christopher Longuet-Higgins and Kvetoslav Prazdny. The interpretation of a moving retinal image. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 208(1173):385–397, 1980.

[104] Weng Fei Low, Zhi Gao, Cheng Xiang, and Bharath Ramesh. Sofea: A non-iterative and robust optical flow estimation algorithm for dynamic vision sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 82–83, 2020.

[105] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[106] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[107] Bruce David Lucas. Generalized image matching by the method of differences. 1985.

[108] Iulia-Alexandra Lungu, Federico Corradi, and Tobi Delbrück. Live demonstration: Convolutional neural network driven by dynamic vision sensor playing roshambo. In *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*, pages 1–1. IEEE, 2017.

[109] W James MacLean, Allan D Jepson, and Richard C Frecker. Recovery of egomotion and segmentation of independent object motion using the em algorithm. In *BMVC*, pages 1–10, 1994.

[110] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso Garcıa, 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.

[111] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. *arXiv preprint arXiv:1711.07837*, 2017.

[112] Paul A Merolla, John V Arthur, Rodrigo Alvarez-Icaza, Andrew S Cassidy, Jun Sawada, Filipp Akopyan, Bryan L Jackson, Nabil Imam, Chen Guo, Yutaka Nakamura, et al. A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, 345(6197):668–673, 2014.

[113] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[114] Diederik Paul Moeys, Federico Corradi, Emmett Kerr, Philip Vance, Gautham Das, Daniel Neil, Dermot Kerr, and Tobi Delbrück. Steering a predator robot using a mixed frame/event-driven convolutional neural network. In *Event-based Control, Communication, and Signal Processing (EBCCSP), 2016 Second International Conference on*, pages 1–8. IEEE, 2016.

[115] Elias Mueggler, Nathan Baumli, Flavio Fontana, and Davide Scaramuzza. Towards evasive maneuvers with quadrotors using dynamic vision sensors. In *Mobile Robots (ECMR), 2015 European Conference on*, pages 1–8. IEEE, 2015.

[116] Elias Mueggler, Christian Forster, Nathan Baumli, Guillermo Gallego, and Davide Scaramuzza. Lifetime estimation of events from dynamic vision sensors. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 4874–4881. IEEE, 2015.

[117] Elias Mueggler, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. Continuous-time visual-inertial odometry for event cameras. *IEEE Transactions on Robotics*, 34(6):1425–1440, 2018.

[118] Elias Mueggler, Guillermo Gallego, and Davide Scaramuzza. Continuous-time trajectory estimation for event-based vision sensors. Technical report, 2015.

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

[120] 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.

[121] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *VISAPP (1)*, 2(331-340):2, 2009.

[122] Jun Nagata, Yusuke Sekikawa, Kosuke Hara, and Yoshimitsu Aoki. Foe-based regularization for optical flow estimation from an in-vehicle event camera. *Electronics and Communications in Japan*, 103(1-4):19–25, 2020.

[123] Hans-Hellmut Nagel. Displacement vectors derived from second-order intensity variations in image sequences. *Computer Vision, Graphics, and Image Processing*, 21(1):85–117, 1983.

[124] Hans-Hellmut Nagel. On the estimation of optical flow: Relations between different approaches and some new results. *Artificial intelligence*, 33(3):299–324, 1987.

[125] Hans-Hellmut Nagel and Wilfried Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):565–593, 1986.

[126] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. Phased lstm: Accelerating recurrent network training for long or event-based sequences. In *Advances in Neural Information Processing Systems*, pages 3882–3890, 2016.

[127] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie

Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. IEEE, 2011.

[128] Anh Nguyen, Thanh-Toan Do, Darwin G Caldwell, and Nikos G Tsagarakis. Real-time 6dof pose relocalization for event cameras with stacked spatial lstm networks. *arXiv preprint*.

[129] Z Ni, C Pacoret, R Benosman, S Ieng, et al. Asynchronous event-based high speed vision for microparticle tracking. *Journal of microscopy*, 245(3):236–244, 2012.

[130] Zhenjiang Ni, Aude Bolopion, Joël Agnus, Ryad Benosman, and Stéphane Régnier. Asynchronous event-based visual shape tracking for stable haptic feedback in microrobotics. *IEEE Transactions on Robotics*, 28(5):1081–1089, 2012.

[131] Zhenjiang Ni, Cécile Pacoret, Ryad Benosman, Siohoi Ieng, and Stéphane RÉGNIER*. Asynchronous event-based high speed vision for microparticle tracking. *Journal of microscopy*, 245(3):236–244, 2012.

[132] Urbano Miguel Nunes and Yiannis Demiris. Entropy minimisation framework for event-based vision model estimation. In *European Conference on Computer Vision*, pages 161–176. Springer, 2020.

[133] Garrick Orchard, Ryad Benosman, Ralph Etienne-Cummings, and Nitish V Thakor. A spiking neural network architecture for visual motion estimation. In *Biomedical Circuits and Systems Conference (BioCAS), 2013 IEEE*, pages 298–301. IEEE, 2013.

[134] 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.

[135] 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.

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

[137] Paul KJ Park, Baek Hwan Cho, Jin Man Park, Kyoobin Lee, Ha Young Kim, Hyo Ah Kang, Hyun Goo Lee, Jooyeon Woo, Yohan Roh, Won Jo Lee, et al. Performance improvement of deep learning based gesture recognition using spatiotemporal demosaicing technique. In *Image Processing (ICIP), 2016 IEEE International Conference on*, pages 1624–1628. IEEE, 2016.

[138] Xin Peng, Yifu Wang, Ling Gao, and Laurent Kneip. Globally-optimal event camera motion estimation. In *European Conference on Computer Vision*, pages 51–67. Springer, 2020.

[139] Xin Peng, Yifu Wang, Ling Gao, and Laurent Kneip. Globally-optimal event camera motion estimation. In *European Conference on Computer Vision*, pages 51–67. Springer, 2020.

[140] Ewa Piatkowska, Ahmed Nabil Belbachir, Stephan Schraml, and Margrit Gelautz. Spatiotemporal multiple persons tracking using dynamic vision sensor. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 35–40. IEEE, 2012.

[141] Christoph Posch, Daniel Matolin, and Rainer Wohlgenannt. A qvga 143 db dynamic range frame-free pwm image sensor with lossless pixel-level video compression and time-domain cds. *IEEE Journal of Solid-State Circuits*, 46(1):259–275, 2010.

[142] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, page 2. IEEE, 2017.

[143] Henri Rebecq, Timo Horstschäfer, Guillermo Gallego, and Davide Scaramuzza. Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time. *IEEE Robotics and Automation Letters*, 2(2):593–600, 2016.

[144] Christian Reinbacher, Gottfried Graber, and Thomas Pock. Real-time intensity-image reconstruction for event cameras using manifold regularisation. *arXiv preprint arXiv:1607.06283*, 2016.

[145] Christian Reinbacher, Gottfried Munda, and Thomas Pock. Real-time panoramic tracking for event cameras. In *2017 IEEE International Conference on Computational Photography (ICCP)*, pages 1–9. IEEE, 2017.

[146] Jerome Revaud, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1164–1172, 2015.

[147] David Reverter Valeiras, Garrick Orchard, Sio-Hoi Ieng, and Ryad B Benosman. Neuromorphic event-based 3d pose estimation. *Frontiers in neuroscience*, 9:522, 2016.

[148] 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.

[149] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533, 1986.

[150] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Continuous-time intensity estimation using event cameras. In *Asian Conference on Computer Vision*, pages 308–324. Springer, 2018.

[151] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Continuous-time intensity estimation using event cameras. In *Asian Conference on Computer Vision*, pages 308–324. Springer, 2018.

[152] Cedric Scheerlinck, Nick Barnes, and Robert Mahony. Asynchronous spatial image convolutions for event cameras. *IEEE Robotics and Automation Letters*, 4(2):816–822, 2019.

[153] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[154] Xavier Lagorce Cédric Meyer Sio-Hoi and Ieng David Filliat Ryad Benosman. Asynchronous event-based multi-kernel algorithm for high speed visual features tracking.

[155] 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.

[156] AB Stanbridge and DJ Ewins. Modal testing using a scanning laser doppler vibrometer. *Mechanical Systems and Signal Processing*, 13(2):255–270, 1999.

[157] Timo Stoffregen and Lindsay Kleeman. Simultaneous optical flow and segmentation (sofas) using dynamic vision sensor. *arXiv preprint arXiv:1805.12326*, 2018.

[158] Timo Stoffregen and Lindsay Kleeman. Event cameras, contrast maximization and reward functions: an analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12300–12308, 2019.

[159] Keith Sullivan and Wallace Lawson. Representing motion information from event-based cameras. In *Robot and Human Interactive Communication (RO-MAN), 2017 26th IEEE International Symposium on*, pages 1465–1470. IEEE, 2017.

[160] Deqing Sun, Stefan Roth, and Michael J Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision*, 106(2):115–137, 2014.

[161] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018.

[162] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[163] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[164] David Reverter Valeiras, Xavier Lagorce, Xavier Clady, Chiara Bartolozzi, Sio-Hoi Ieng, and Ryad Benosman. An asynchronous neuromorphic event-driven visual part-based shape tracking. *IEEE transactions on neural networks and learning systems*, 26(12):3045–3059, 2015.

[165] 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.

[166] 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.

[167] Neal Wadhwa, Michael Rubinstein, Frederic Durand, and William T Freeman. Riesz pyramids for fast phase-based video magnification, May 10 2016. US Patent 9,338,331.

[168] Yifu Wang, Jiaqi Yang, Xin Peng, Peng Wu, Ling Gao, Kun Huang, Jiaben Chen, and Laurent Kneip. Visual odometry with an event camera using continuous ray warping and volumetric contrast maximization. *arXiv preprint arXiv:2107.03011*, 2021.

[169] Christopher Warren, Christopher Niezrecki, Peter Avitabile, and Pawan Pingle. Comparison of frf measurements and mode shapes determined using optically image based, laser, and accelerometer measurements. *Mechanical Systems and Signal Processing*, 25(6):2191–2202, 2011.

[170] David Weikersdorfer, David B Adrian, Daniel Cremers, and Jörg Conradt. Event-based 3d slam with a depth-augmented dynamic vision

sensor. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 359–364. IEEE, 2014.

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

[172] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1385–1392, 2013.

[173] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frédo Durand, and William Freeman. Eulerian video magnification for revealing subtle changes in the world. 2012.

[174] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[175] Jie Xu, Meng Jiang, Lei Yu, Wen Yang, and Wenwei Wang. Robust motion compensation for event cameras with smooth constraint. *IEEE Transactions on Computational Imaging*, 6:604–614, 2020.

[176] Jie Xu, Meng Jiang, Lei Yu, Wen Yang, and Wenwei Wang. Robust motion compensation for event cameras with smooth constraint. *IEEE Transactions on Computational Imaging*, 6:604–614, 2020.

[177] Yongchao Yang, Charles Dorn, Tyler Mancini, Zachary Talken, Garrett Kenyon, Charles Farrar, and David Mascareñas. Blind identification of full-field vibration modes from video measurements with phase-based video motion magnification. *Mechanical Systems and Signal Processing*, 85:567–590, 2017.

[178] Chengxi Ye, Anton Mitrokhin, Chethan Parameshwara, Cornelia Fermüller, James A Yorke, and Yiannis Aloimonos. Unsupervised learning of dense optical flow and depth from sparse event data. *arXiv preprint arXiv:1809.08625*, 2018.

[179] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based feature tracking with probabilistic data association. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4465–4470. IEEE, 2017.

[180] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Ev-flownet: Self-supervised optical flow estimation for event-based cameras. *arXiv preprint arXiv:1802.06898*, 2018.

[181] Alex Zihao Zhu, Liangzhe Yuan, Kenneth Chaney, and Kostas Daniilidis. Unsupervised event-based learning of optical flow, depth, and egomotion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 989–997, 2019.

[182] Alex Zihao Zhu, Nikolay Atanasov, and Kostas Daniilidis. Event-based visual inertial odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5391–5399, 2017.