Estimating Motion with an Event Camera

July 2023

Shintaro Shiba

A THESIS FOR THE DEGREE OF

PH.D.IN ENGINEERING

Estimating Motion with an Event Camera

JULY 2023

Keio University

GRADUATE SCHOOL OF SCIENCE AND TECHNOLOGY
KEIO UNIVERSITY

Shintaro Shiba

Abstract

Estimating motion from image sensors is a fundamental problem in computer vision and robotics. Event cameras are novel bio-inspired sensors that provide a signal suitable for estimating motion because their pixels naturally respond to intensity changes, which are produced by moving patterns on the image plane. This working principle of visual data acquisition in the form of asynchronous per-pixel intensity differences offers advantages such as low latency, high dynamic range, and data efficiency, which contribute to overcoming challenging scenarios for conventional frame-based cameras. Motion estimation as a fitting problem can be categorized according to the hypothesis of the scene, such as globally uniform flow (2 degrees of freedom (DOFs)), homographic-based flow (8 DOFs), or optical flow (the highest number of DOFs). However, state-of-the-art methods of event-based motion estimation have several challenges: (*i*) typical objective functions have undesired global/local optima for complex ego-motion and optical flow estimation scenarios, (*ii*) event-based optical flow has not considered the space-time nature of events, and (*iii*) many existing optical flow methods are not biologically plausible. This thesis rethinks the nature of event data, improves the well-posedness of various motion estimation problems, and proposes a new optical flow estimation approach. Furthermore, it demonstrates a novel application of event-based motion estimation in imaging sciences.

The thesis is organized as follows: Chapter 1 gives an overview of the motion estimation problems considered using an event camera, summarizing the contributions of this work. Chapter 2 reviews the working principle of an event camera and existing methods in motion estimation that use only events or a combination of events and frames (i.e., images). Chapter 3 focuses on low-DOF ego-motion estimation from events alone and proposes improvements to the Contrast Maximization (CMax) framework. The goal of this chapter is to extend it to higher, more complex motion estimation problems by mitigating event collapse without trading off speed. Chapter 4 focuses on high-DOF optical flow estimation and proposes a principled method to estimate optical flow by extending CMax. It also extends frame-based optical flow to event-based, space-time optical flow to handle occlusions better. Chapter 5 proposes a new optical flow estimation method, which achieves fast runtime upon sacrificing accuracy. The proposed method stems from neuroscience and is biologically plausible.

Chapter 6 demonstrates an application to estimate the convection of heated air (motion of air density), using schlieren imaging techniques. Here, a new method using events and frames to estimate complex motion is proposed, by extending the linearized event generation model. Chapter 7 summarizes the results of this work and discusses future work.

This thesis deepens the understanding of various motion estimation tasks using CMax, rethinks the space-time nature of the data in event-based optical flow, highlights the speed-accuracy trade-off in existing optical flow estimation methods, and pioneers another stack of applications using event-based motion estimation in imaging sciences.

概要

カメラを用いた動きの推定は，コンピュータビジョンやロボティクスにおける基本的な問題の一つである．生物に着想を得た新しいセンサであるイベントカメラは，画像平面上のエッジの動きなどによって引き起こされる画素の輝度の変化に反応するため，動きの推定に適したデータを得ることができる．このような輝度の差分データを非同期的で時空間的に取得する原理によって，低遅延，高ダイナミックレンジ，データ効率性などの利点をもつため，イベントカメラは従来のフレームカメラでは困難だった環境での活躍が期待されている．動き推定は，与えられた入力データと動きの仮説（モデル）に対するパラメータの推定問題であり，例えば画像全体で均一な動き（2 自由度），ホモグラフィ変換（8 自由度），オプティカルフローなど，推定関数の複雑さに沿って分類できる．しかしながら，イベントベースで動き推定をおこなう最先端の手法には，いくつかの課題がある：(*i*) 典型的な目的関数は，複雑な自己動き（エゴモーション）やオプティカルフローの推定において，望ましくない最適解を持つため，推定が不安定である．(*ii*) イベントベースのオプティカルフロー推定は，イベントの時空間的な性質を考慮してきていない．(*iii*) 既存のオプティカルフロー手法の多くは，生物学的に妥当であるとは言い難い．本研究では，イベントデータの性質を再考することで，自己動きやオプティカルフローといった様々な動き推定問題における不良な設定を改善し，新しいオプティカルフロー推定手法を提案する．さらに，イベントベースの動き推定の応用として，新たなイメージング技術を開発する．

第 1 章では，イベントカメラを用いた動き推定問題の概要を提示し，本研究の貢献についてまとめる．第 2 章では，イベントカメラの原理と，イベントカメラを用いた様々な動き推定に関する既存の手法を概観する．第 3 章では，低自由度の自己動き推定問題に注目し，イベントのみから動きを推定するコントラスト最大化法の改良を提案する．この章の目標は，速度を犠牲にすることなくイベント崩壊を緩和することで，より高度で複雑な自己動き推定問題を安定的に解くことである．第 4 章は，高自由度のオプティカルフロー推定問題に焦点を当て，コントラスト最大化法を拡張することでオプティカルフローを推定する原理的な方法を提案する．ここでは，オプティカルフローの定

義をフレームベースからイベントベースの時空間オプティカルフローに拡張し，遮蔽をよりうまく扱えるようにしている．第 5 章では，神経科学の知見に基づいた，生物学的により妥当な新しいオプティカルフロー推定手法を提案し，第 4 章の手法とも比較しながら，実行速度と推定精度のトレードオフについて議論する．第 6 章では，動き推定の応用として，イベントカメラを使用したシュリーレンイメージング技術を開発し，空気の対流を推定する手法を提案する．ここでは，第 4 章で提案したフロー推定手法とも比較しながら，線形化イベント生成モデルを拡張し，イベントとフレームを用いてシュリーレンのような複雑な運動を推定する新しい手法を提案する．第 7 章では，本研究の成果をまとめ，今後の課題について議論する．

本論文は，コントラスト最大化を用いた様々な動き推定への理解を深め，イベントベースのオプティカルフローにおけるデータの時空間的な性質の再考を促し，既存のオプティカルフロー推定手法における速度と精度のトレードオフについて明らかにし，さらに，イベントカメラを用いた新しいイメージングアプリケーションを開拓する．

Acknowledgments

This research was conducted under the supervision of Prof. Dr. Yoshimitsu Aoki, Graduate School of Science and Technology, Keio University, while I was a doctoral student at the Graduate School of Science and Technology, Keio University. Completing this Ph.D. thesis would not have been possible without the support, guidance, and encouragement of numerous individuals who have played significant roles.

First and foremost, I express my sincere gratitude to my supervisor, Prof. Dr. Yoshimitsu Aoki, who was the main supervisor of this thesis and my advisor. Since Prof. Aoki accepted me, who had finished his Master's in neuroscience, into the doctoral program in engineering, he has given me warm and constructive guidance anytime in good progress or not during the doctoral research. His intellectual stimulation and mentorship have been pivotal in shaping my research skills and critical thinking abilities. He also supported me in various ways not only in the first half of the doctoral course, which began as a working-doctoral course, but also in the second half, which led to my stay in Germany.

I would also like to express my sincere gratitude to Prof. Dr. Guillermo Gallego, the Department of Electrical Engineering and Computer Science (Faculty IV) at Technische Universität Berlin, for his unwavering support, valuable insights, and constructive feedback as the co-supervisor and also as a mentor. Prof. Gallego, a pioneer in the field of event-based vision, taught me not only a deep understanding and technical excellence in the research topics, but also the way of tackling truly challenging problems, and the attitude towards principled and essential contributions to the field. Once again, I would like to express my sincere and deepest gratitude to both of my Ph.D. supervisors for their guidance.

My appreciation also goes to my friends and colleagues in Aoki Media Laboratory and the TUB Robotics Interactive Perception group, who made my journey much more manageable, enjoyable, and memorable. It was a great opportunity for me to collaborate through their expertise and to deepen my understanding of the research project from their various viewpoints. I would also like to express my appreciation to the secretaries of both groups, and the network admin team in Aoki Media Lab, who maintained the GPU servers.

I would also like to thank Prof. Dr. Kostas Daniilidis, the University of Pennsylvania, Prof. Dr. Masaaki Ikehara, and Prof. Dr. Hideo Saito, Keio University, who kindly agreed

to be part of the thesis committee, for their valuable comments and suggestions during the proposal and final defense. Their expertise and constructive feedback were instrumental in shaping the outcome of this thesis. In addition to the content of my research, I have also learned the attitude toward research, scientific writing, and communicating in the presentation. I want to express my gratitude again to all the thesis reviewers.

I would also like to thank members of Woven Planet Holdings, Inc. for supporting me to enter the doctoral program. In particular, I would like to thank Dr. Yusuke Yachide and the human resources team for their efforts to allow me to do my working-doctoral program and then take a leave of absence to study abroad. Dr. Yachide not only supported me with the administrative procedures, but also helped me shape the research proposal to begin my doctoral studies.

Last, but not least, I am deeply grateful to my family. In particular, I would like to thank my mom. She has been consistently positive and supportive, despite we lost my dad right after I had started my doctoral program, resulting in great anxiety, both mentally and financially. I would also like to thank my other family members for their unwavering love, encouragement, and belief in my potential which have been the driving force behind my success. Finally, I express my sincere gratitude to my partner, Ms. Anna Ebe, who has been a constant source of enormous support throughout my academic journey.

謝辞

本研究は，著者が慶應義塾大学大学院理工学研究科後期博士課程在学中に，同大学理工学部青木義満教授の指導のもとおこなわれました．本論文の執筆にあたり，多くの方からご指導，ご支援をいただいたことに感謝します．

はじめに，本論文の主査であり，指導教員である青木義満教授に心より感謝申し上げます．青木義満教授には，生物系の修士課程を卒業した身である筆者を暖かく工学の博士課程に受け入れてくださったところから，研究がうまく進まないときもうまく進んでいるときも，変わらず温かく建設的なご指導をいただきました．また，社会人博士課程として始まった博士課程前半だけでなく，ドイツ留学に至った後半においても様々な形で支援していただきました．そのような指導や支援なしでは博士課程の終了が成し得なかったであろうことは想像に難くありません．

また，留学中に指導して下さったベルリン工科大学電気工学・コンピュータ科学学科（Faculty IV）Gallego Guillermo 教授にも心より感謝申し上げます．イベントカメラという先駆的な分野において実質的な開拓者である Gallego 教授からは，研究分野の深い理解だけでなく，困難な問題に正面からアプローチし，原理的で本質的な手法の開発にこだわる姿勢を学ばせていただきました．改めて指導に携わってくださったお二人に心より感謝申し上げます．

また，青木研究室のメンバー，Gallego 研究室で共同研究や議論をおこなったメンバー、そして秘書の方々にも深く感謝します．様々な視点から研究対象への理解を深めるきっかけとなり，研究を通じて切磋琢磨できたことは非常にありがたいことでした．青木研究室において GPU の整備や日々のメンテナンスに貢献してくださったネットワーク係にも感謝を申し上げたいです．

本論文の審査にあたり，快く副査を引き受けてくださった，ペンシルバニア大学工学部の Daniilidis Kostas 教授，慶應義塾大学理工学部の池原雅章教授，斎藤秀雄教授にも感謝いたします．学位論文の審査を通して，研究の内容だけでなく，研究に対する姿勢，論文の書き方や発表の仕方まで，幅広く学ばせていただくことができました．副査にあたってくださったことを非常に光栄に思い，博士取得後も引き続き活躍していかなければと身の引き締まる思いです．副査の先生方に改めて感謝申し上げます．

さらに，博士課程進学にあたって許可をしてくださった株式会社 Woven Planet Holdings の皆様にも感謝いたします．特に，社会人博士をおこない，またその後休職し留学するにあたって，許可に向けて動いてくださった谷内出悠介博士と人事の皆様に感謝します．谷内出博士には，手続き上の支援だけでなく，博士研究計画の内容にも相談に乗っていただきました．

最後に，私の家族に深く感謝します．特に母には，博士を始めたばかりの頃に父を亡くし，家庭に大きな不安のある中でも，博士課程に向けて前向きで献身的な支援をいただきました．他の家族の皆様も含めて，あらためて感謝申し上げます．最後に，博士課程を通して支えてくださった，パートナーである江部杏奈氏にも深く感謝します．

Contents

Listing of figures

List of Tables

1 Introduction

What is motion and how can we estimate it? This is a fundamental research question in computer vision and robotics. Throughout the thesis, I tackle this problem using an event camera, which is a novel bio-inspired vision sensor. This chapter starts by defining the motion estimation problem using vision sensors, in Sec. 1.1. Secondly, Sec. 1.2 introduces event cameras and the challenges in event-based motion estimation. Then, the contributions and structure of this thesis are summarized in Sec. 1.3. Finally, after providing the list of publications in Sec. 1.4, Sec. 1.5 concludes the chapter.

1.1 Motion Estimation

Estimating the motion of the world within short time intervals is a challenging task for computers and robots. Animals perceive their motion and the motion of their surroundings instantly and precisely to survive, communicate, or migrate. We humans also recognize motion, heavily relying on vision (i.e., eyes). Analogously, for decades researchers have developed computer systems that recognize motion using visual sensors (i.e., cameras). The application of vision-based motion estimation varies widely across fields, such as tracking, simultaneous-localization and mapping (SLAM), controlling autonomous robots, scene prediction, video synthesis, sports analysis, or augmented reality.

Motion estimation can be defined as a fitting problem. A fitting problem needs (*i*) in-

put data, (*ii*) function(s), and (*iii*) parameter(s) to fit. For example, for traditional CMOS- or CCD-based imaging sensors, which are *frame-based* cameras, the input data is an image (frame) or a sequence of frames (video). The fitting function comes from modeling the world. Since we do not know the best functions that universally and efficiently represent any motion in the world, we need to make some assumptions to define the fitting function, which involves modeling (i.e., approximation) of reality. For example, if we model the world as flat and stationary, the fitting function becomes a homography transformation due to the ego-motion. If one wants to describe as complex motions as possible in an image plane, the fitting function can be the per-pixel displacement in the image plane, called optical flow. The motion hypothesis, in this case, is that each pixel displacement is linear; optical flow between two consecutive frames is a function of the pixel location, but not of time. The choice of the motion hypothesis affects the complexity of the problem, such as the number of parameters (*degrees of freedom*: DOFs). It is also closely related to the downstream application (i.e., *why* does one estimate motion?) and to the problem's difficulty (i.e., *how* can the problem be solved more easily while remaining useful?).

Finally, an estimation method seeks the best parameters for the input data and the function. One approach to finding them is by solving an optimization problem, where the objective function plays an important role. For example, the reprojection error that measures distances between the points in two images can be used to estimate the homography transformation between two consecutive frames. For better convergence of the optimization solver, it is important to design the objective function so that it has a good landscape (i.e., convexity) and few local sub-optima, ideally, a single global optimum that reflects reality. The combination of the motion hypothesis and objective function affects the *well-posedness* of the problem, which determines the stability of the convergence, the dependency on the initial conditions, and the proneness to overfitting. Another estimation method is *learning-based*, such as deep neural networks (DNN). Although the training in the learning-based methods also involves the optimization process, the parameters to be optimized are not the motion parameters themselves, but the weights and biases of the network that will be used to output the motion parameters. This design difference makes them more data-driven, since the network architecture partly encapsulates the design of the fitting function and the training process automates finding the best function by exploiting the statistical correlation in the input data.

This thesis tackles the various motion estimation problems from an event camera (Fig. 1.1). Specifically, the problem settings of interest are as follows: (*i*) the input data considered are *events* or *events and a frame*, (*ii*) the motion hypotheses considered are *egomotion from 1 DOF to 8 DOFs* and *optical flow*, and (*iii*) the estimation methods considered are *optimization-based* and *learning-based*. The downstream applications, limitations due to the approximation of each motion hypothesis, accuracy, and computational complexity are partly discussed in each chapter and will be revisited in the final discussion.

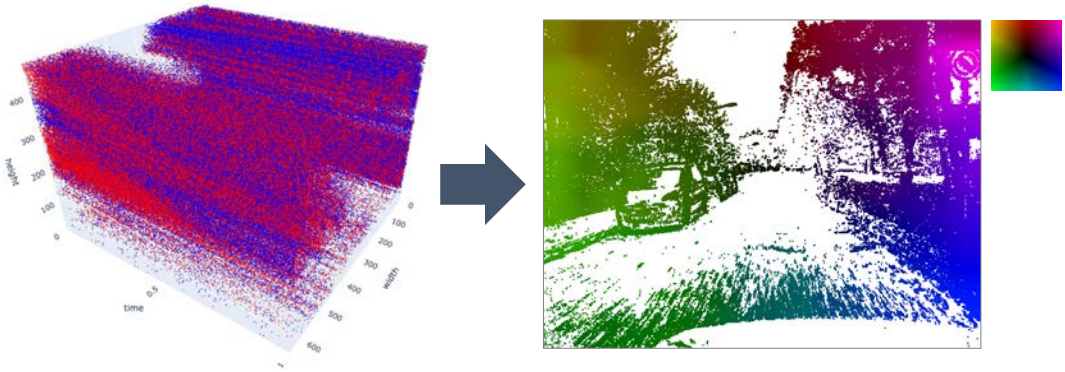


Figure 1.1: *Motion estimation from an event camera.* In this example, input data is a stream of events from an event camera (see also: Fig. 2.1), the motion model is optical flow (discussed in Chapters 4 and 5), and the estimation method is optimization-based. The color denotes the flow direction and magnitude (see the color wheel). This is an example of the results from Chapter 4.

Here, we formulate the problems as purely visual ones, which do not require input from additional sources such as an inertial measurement unit (IMU) or a light detection and ranging (LiDAR) sensor. Also, we limit the scope of the motion estimation problem such that it takes input data only from nearby timestamps (short time intervals): motion estimation in longer time intervals, such as tracking and global bundle adjustment (e.g., SLAM), are future directions in which this work could be extended.

1.2 Motion and Event Cameras

Event cameras are novel bio-inspired vision sensors that respond to per-pixel intensity changes. Assuming constant illumination, the intensity change is caused by the relative motion of edges in image space, and therefore events naturally provide a signal suitable for motion estimation. Compared with frame-based cameras, event cameras offer advantages, such as high dynamic range (HDR), data and power efficiency, high temporal resolution (on the order of μs), and minimal motion blur. These advantages make them useful for accurate motion estimation even in difficult real-world scenarios for frame-based cameras. Another aspect is the analogy of its data-acquisition principles to the animal retina, which attracts researchers not only from computer science but from robotics and neuroscience communities.

Contrast maximization (CMax)[46,45] is an optimization-based framework that provides state-of-the-art results on motion-related tasks, such as rotational motion estimation[47], feature flow estimation and tracking, 3D reconstruction, and optical flow estimation. The main

idea of CMax and similar event alignment frameworks is to find the motion and/or scene parameters to maximize an objective function (e.g., contrast) that measures the alignment of corresponding events caused by the same scene edge. However, the contrast functions in the CMax have undesired solutions (global optimum or local optima) where events accumulate into too few pixels in some cases, such as optical flow and some low-DOF ego-motion estimation. Prior works have largely ignored the issue or proposed workarounds without analyzing the phenomenon in detail. Since CMax is at the heart of many state-of-the-art methods, it is important to understand this phenomenon and propose new methods that extend CMax for broader, complex motion hypotheses. We analyze this phenomenon, which we call *event collapse*, show that it occurs according to the well-posedness of the problem, and propose solutions that improve the well-posedness of motion estimation from events alone.

Another challenge in event-based motion estimation is its new data modality. The asynchronous and sparse events from these cameras happen, by nature, in *space-time*, which is not compatible with traditional computer vision algorithms designed for images. Optical flow is no longer obtained by analyzing the intensities of images captured at two nearby timestamps, but by analyzing the space-time stream of events. This leads us to rethink visual processing and demands new algorithms that are suitable for event-based optical flow. Specifically, we focus on rethinking optical flow in event-based vision such that the space-time nature of events is taken into account. We propose *space-time flow*, which is a function of space-time and is constant along the streamlines of the flow itself, thus handling occlusions better.

Furthermore, we explore a more biologically-plausible optical flow estimation method for the event-based optical flow. Most existing optical flow estimation methods for events are batch-based, taking input data as a packet of events on a fixed time interval (e.g., 10–100 ms) or with a fixed number (e.g., 30k–1M). While they generally achieve good estimation accuracy, these packet-based methods require some waiting time before the processing (inference) starts: they trade off the high-speed advantages of event data for accuracy. On the other hand, *event-by-event* methods process every event incrementally as it occurs (without waiting time), aiming to leverage the camera's low-latency advantage. As a counterpart of the other (batch-based) approaches, we propose an event-by-event sparse optical flow estimation method, stemming insights from neuroscience, and demonstrate its capability for fast execution ($>$ 10 kHz rate) on standard CPUs.

Finally, we explore another stack of applications in imaging sciences that utilize event-based motion estimation: the motion of air density. Schlieren imaging is an optical technique to observe the flow of transparent media, such as air or water, without any particle seeding. However, conventional frame-based schlieren techniques require both high spatial and temporal resolution cameras, which impose bright illumination and expensive computation limitations. Event cameras can overcome such limitations of frame-based imaging techniques due to their bio-inspired sensing principles. We pioneer a novel technique for perceiving air

convection (i.e., motion) using events and frames, which leverages the advantages of event cameras.

To summarize, in this work we focus on motion estimation problems within short time intervals using a single event camera and hope to challenge the following research questions:

- How can we extend CMax for broader types of motion hypotheses by improving the objective function?
- How can we take space-time nature of events into account to rethink event-based optical flow?
- What is a more biologically-plausible solution for event-based optical flow?
- How can we utilize event-based motion estimation in imaging science, in order to leverage its advantages?

On the other hand, events could be also triggered by illumination changes, such as flickering lights and noise, which do not carry information of moving objects in the scene or ego-motion. Although this is another interesting challenge, we limit the scope of this work by excluding these non-constant illumination situations. Note that there are also other challenges that are out of the scope of this thesis, such as the integration with embedded hardware and achieving power-efficient systems during motion estimation.

1.3 Contributions

The contributions of this thesis are summarized as follows:

- We provide the first detailed analysis of event collapse that occurs at specific motion estimation problems in the CMax due to undesired optima of the objective functions (Chapter 3). The well-posedness of the motion hypothesis is discussed for low-DOF motions (1 to 8 DOFs).
- We propose three regularizers to improve the well-posedness in the low-DOF motion estimation problems (Chapter 3). The proposed regularizers are the only effective approach to date to mitigate event collapse in these settings, one of which even does not sacrifice the runtime of the original CMax framework.
- We propose a multi-reference warp and focus loss to drastically improve optical flow accuracy by discouraging event collapse (Chapter 4). By extending the CMax, the proposed method achieves state-of-the-art accuracy in event-based optical flow.

Table 1.1: The problem settings in each chapter of the thesis.

Chapter	Input data	Estimation method	Motion hypothesis
3	events	optimization	low DOF, ego-motion
4	events	optimization and learning	high DOF, optical flow
5	events	optimization (event-by-event)	high DOF, optical flow
6 (imaging application)	events and a frame	optimization	high DOF, optical flow

- We propose a principled time-aware flow to comply with the space-time nature of events, which handles occlusions better (Chapter 4). The space-time flow is formulated as a transport problem via partial differential equations (PDEs).
- We show a multi-scale approach on the raw events to improve the convergence to the solution and avoid getting trapped in local optima in event-based optical flow (Chapter 4).
- We propose a biologically-plausible method to estimate optical flow based in an event-by-event manner, leveraging knowledge from neuroscience. The proposed method achieves the fastest runtime in optical flow benchmarks (Chapter 5).
- We demonstrate a novel imaging application of event cameras with schlieren techniques (Chapter 6). We develop a theory to connect events and heat convection in air, then estimate the temporal change of air density using an event camera.
- We provide a new dataset with high-quality frames and events for event-based background-oriented schlieren (Chapter 6).

A thorough evaluation on the de-facto standard benchmarks and datasets is conducted in each problem setting. Also, the runtime and accuracy among the proposed methods are compared for low-DOF motion estimation (Chapter 3) and optical-flow estimation (Chapter 5). Finally, most parts of the thesis provide publicly available implementations (codes) for the future of the event-vision community.

This thesis consists of the following chapters (Tab. 1.1):

- Chapter 1 defines the scope of the thesis and the research questions.
- Chapter 2 reviews and summarizes existing work on motion estimation using an event camera.
- Chapter 3 focuses on low-DOF estimation problems using only events. Here, the motion hypotheses are limited to up to 8 DOFs due to the camera ego-motion. We analyze

event collapse in detail and propose a new method with new regularizers that effectively and/or efficiently improve the accuracy. The estimation methods are optimization-based ones.

- Chapter 4 focuses on the high-DOF estimation problem (optical flow) using only events. We propose a principled method to estimate optical flow by extending Contrast Maximization. The estimation methods are both optimization-based (CMax) and learning-based (DNN).

- Chapter 5 also focuses on optical flow estimation using events, by proposing an incremental approach. The proposed method performs event-by-event estimation, thus achieving high throughput (runtime per event).

- Chapter 6 shows a novel application of event-based motion estimation: sensing air convection. In contrast to the previous chapters, here the task is estimating the motion of air density using schlieren techniques. The input data is the combination of events and a frame. We estimate the spatio-temporal derivatives of air density via optical flow estimation. The estimation method is based on optimization, where we extend the linearized event generation model.

- Chapter 7 summarizes the work, discusses the limitations, and provides an outlook.

1.4 Publication list

The publication list of the author and the corresponding chapters in the thesis are as follows:

- Shiba, S., Aoki, Y., & Gallego, G. (2022). Event Collapse in Contrast Maximization Frameworks. *Sensors*, 22(14):5190, doi: 10.3390/s22145190 (Chapter 3)

- Shiba, S., Aoki, Y., & Gallego, G. (2023). A Fast Geometric Regularizer to Mitigate Event Collapse in the Contrast Maximization Framework. *Advanced Intelligent Systems*, 2200251, doi: 10.1002/aisy.202200251 (Chapter 3)

- Shiba, S., Aoki, Y., & Gallego, G. (2022). Secrets of Event-based Optical Flow. In *Eur. Conf. Comput. Vis. (ECCV)*, Tel Aviv, Israel, doi: 10.1007/978-3-031-19797-0_36 (Chapter 4)

- 芝慎太朗, 青木義満, & Gallego, Guillermo. (2022). イベントカメラを用いたオプティカルフロー推定: 動きとは何か？. In ビジョン技術の実利用ワークショップ *(*オンライン*)*. (Chapter 4)

- Shiba, S., Klose, Y., Aoki, Y., & Gallego, G. (*under review*). Secrets of Event-based Optical Flow, Depth and Ego-motion Estimation by Contrast Maximization. (Chapter 4)

- Shiba, S., Aoki, Y., & Gallego, G. (2023). Fast Event-based Optical Flow Estimation by Triplet Matching. *IEEE Signal Processing Letters*, vol. 29, pp. 2712-2716, 2022, doi: 10.1109/LSP.2023.3234800. (Chapter 5)

- Shiba, S., Hamann, F., Aoki, Y., & Gallego, G. (*under review*). Event-based Background-Oriented Schlieren. (Chapter 6)
  I (SS) was the sole first author and main contributor to developing the idea, implementation, and paper writing. FH and I equally contributed to data acquisition and experiments.

1.5 Summary

In this chapter, we define the scope of the thesis as motion estimation within short time intervals using a single event camera. The input data in this work are (*i*) events only (Chapters 3 to 5) or (*ii*) events and a frame (Chapter 6). The motion hypothesis of interest are (*i*) low-DOF ego-motion (up to 8 DOFs) (Chapter 3) and (*ii*) high-DOF optical flow (Chapters 4 to 6). To solve motion estimation problems, optimization-based (Chapters 3, 4 and 6), learning-based methods (Chapter 4), and an incremental method (Chapter 5) are used. As an application of the optical flow estimation, we demonstrate the capability of sensing air convection (Chapter 6).

We hope this thesis expands the understanding of event data and deepens event-based motion estimation by tackling its new data modality and challenges. We also hope this work fosters future research and applications for event-based motion estimation.

2 Review

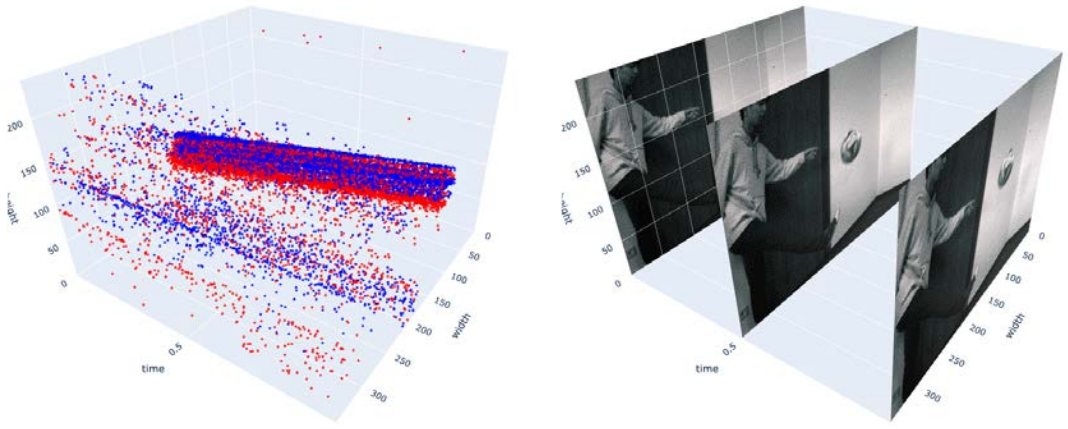
This chapter provides a comprehensive review of event-based motion estimation. First, I summarize the principles of event cameras and their applications in Sec. 2.1. Next, Sec. 2.2 reviews various methods to estimate ego-motion (pose), including the Contrast Maximization framework. Section 2.3 covers optical flow estimation methods. Then, prior work utilizing both events and frames is reviewed in Sec. 2.4. Finally, Sec. 2.5 concludes the chapter.

2.1 Event cameras

Event cameras, or event-based cameras, are novel bio-inspired vision sensors that have the potential to overcome challenging scenarios for traditional frame-based cameras[91,124,13,151]. They respond to intensity changes in the image plane asynchronously, achieving fast and efficient data acquisition (Fig. 2.1). Each pixel of the camera triggers an "event" $e_k \doteq (\mathbf{x}_k, t_k, p_k)$ as soon as its logarithmic intensity L changes from one at the previous event by a certain threshold $C > 0$:

$$L(\mathbf{x}_k, t_k) - L(\mathbf{x}_k, t_k - \Delta t_k) = p_k\, C, \tag{2.1}$$

where $\mathbf{x}_k \doteq (x_k, y_k)^\top$ is the pixel location, t_k is the timestamp, $p_k \in \{+1, -1\}$ is the sign of the intensity change, and $t_k - \Delta t_k$ is the time of the previous event at the same pixel $\mathbf{x}_k$. Compared with conventional frame-based cameras, which synchronously record absolute intensities at a fixed frame rate, event cameras achieve higher temporal resolution and lower latency (in the order of microseconds), higher dynamic range (about 140 dB), and more efficiency in data



(a) Event camera **(b)** Frame Camera

Figure 2.1: *Comparison between the outputs of an event camera and a frame-based camera.* In the scene, a person is playing football. In this scenario the event camera is stationary, thus intensity changes happen only at the pixels of the moving parts (around the football and the human body). The other regions of the image plane (e.g., the door and the wall) are static with respect to the camera, and thus do not trigger events. An event camera (a) outputs asynchronous stream data at only pixels with changes. A frame-based camera (b) outputs a sequence of images at all pixels (synchronously), regardless of the scene dynamics. Here, the camera used is the DAVIS346 [158] (346 $\times$ 260 px).

and power consumption.

Due to these advantages, the applications of event cameras are widely spread from basic computer vision tasks to applications such as robotics and imaging sciences. Classical computer vision tasks include recognition and classification [89,88,115,82], detection [59,99,102], feature tracking [22,86,149,176,54], optical flow estimation [11,114,10,16,6,153,2], image (absolute intensity) reconstruction [79,6,130,118,173], and superresolution [90,101,168,37]. Figure 2.2 shows such an image reconstruction, formulated a linear inverse problem given events and optical flow [173]. They are also used in three-dimensional computer vision, such as depth estimation (3D reconstruction) [128,81,180,76], visual odometry [47,71,94], and Simultaneous Localization and Mapping (SLAM) [79,81,129,135]. Moreover, there are relatively modern computer vision tasks that utilize the high temporal resolution of event cameras, such as image deblurring [92,78] and frame interpolation [14,164]. Recently they have been used in broad applications such as collision avoidance [21,39], structured light 3D scanning [106], particle image velocimetry [35,167], eye tracking [3], surveillance and monitoring [25,24], and autonomous drones [31,39,32]. They are also applied to tasks that require high dynamic range, such as star tracking [20] and HDR image reconstruction [100,181,130].

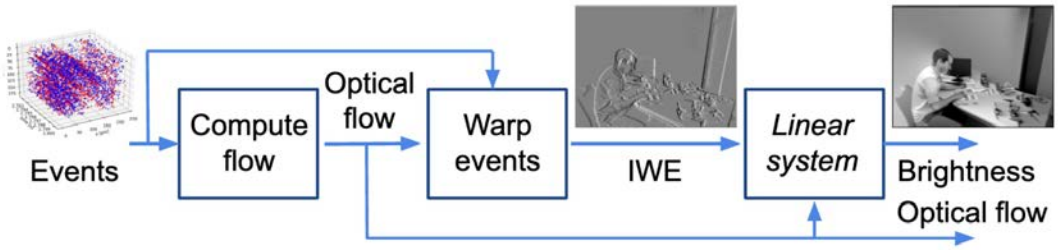


Figure 2.2: *Example of intensity reconstruction.* If optical flow is given, the intensity reconstruction can be formulated as a linear inverse problem. Data from Fig. 1 of [173].

2.2 Motion Estimation from Events: Ego-motion

Since events are caused by the relative motion of edges in image space, motion estimation from event data is an extensive research area in event-based vision. The motion estimation problems are divided into two categories upon an important assumption: if only the camera is moving (and the scene is static), or there might be other moving objects in the scene. This section (Sec. 2.2) describes the first case (*ego-motion estimation*), and the following section (Sec. 2.3) covers the second case (*optical flow estimation*). With the static scene assumption, the "ideal" solution of the motion estimation has $6 + N_p$ DOFs (N_p denotes the number of pixels), where the estimation parameters are the pose change (i.e., velocity) and the scene depth. Due to its complexity, many prior works have posed extra assumptions on the scene or have limited the scope of the problem settings for fewer DOFs as approximations.

The ego-motion estimation problems can be further categorized according to their complexity (degrees of freedom): feature flow (2 DOFs), rotational motion (3 DOFs), planar motion (3 DOFs), similarity transformation (4 DOFs), Affine transformation (6 DOFs), and homography transformation (8 DOFs). For example, the 3-DOF camera rotational motion estimation assumes constant angular velocity ($\theta \equiv \omega$, 3 DOFs) during short time intervals [26,104,132,47,93]. Kim et al. [79] propose to estimate the rotational motion of the camera using Kalman filters, by simultaneously estimating the mosaic image (panorama intensity image). This is extended to estimate the 6-DOF camera pose and the scene depth in [81].

Contrast maximization (CMax) and related event-alignment methods have been used to estimate the motion of various complexity with the current state-of-the-art accuracy [46,113,63]. This optimization-based method iteratively performs transforming (warping) events and computing an objective function from events (Fig. 2.3). The goal is to find the warping (transformation) parameters θ that achieve motion compensation (i.e., alignment of events triggered at different times and pixels), hence revealing the edge structure that caused the events.

Figure 2.4 shows examples for the 2-DOF feature flow ($\theta \equiv \mathbf{v}$. e.g., [46]) and 3-DOF angular

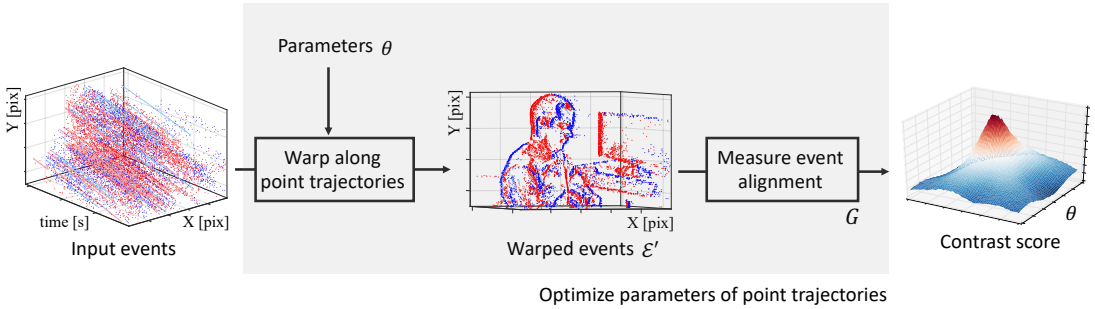


Figure 2.3: *An overview of the Contrast Maximization framework*[46].

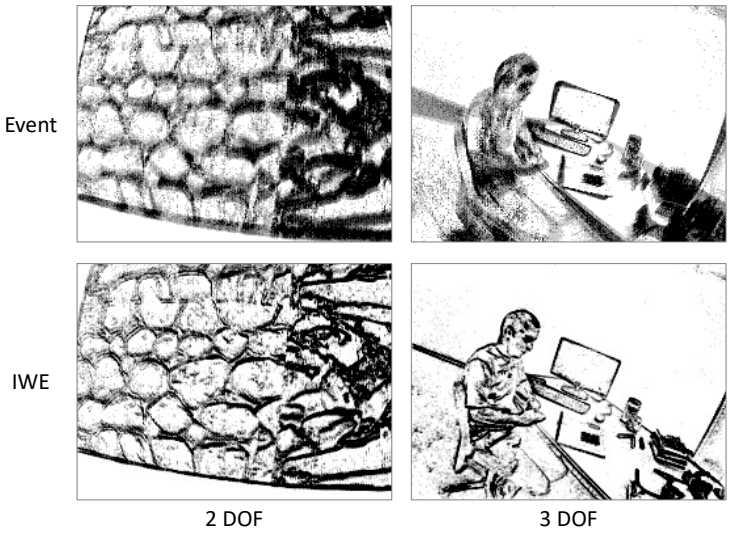


Figure 2.4: *Examples of Contrast Maximization for 2-DOF feature flow and 3-DOF rotational motions.* After convergence, CMax provides the motion parameters and the sharp image of warped events (IWE).

velocity ($\theta \equiv \omega$. e.g.,[47,80]) estimation using CMax. Standard optimization algorithms (gradient ascent, sampling, etc.) can be used to maximize the event-alignment objective functions. Upon convergence, the method provides the best transformation parameters and the transformed events, i.e., a motion-compensated image of warped events (IWE). Several objective functions for measuring event alignment have been proposed to measure the goodness of fit between the events and the model[45,154], which are interpreted as the visual contrast (sharpness) of the IWE.

Applying CMax to more complex (higher DOFs) motions is yet to be explored and investigated. Peng et al.[122,121] analyze the convexity of some contrast functions (from[154]) and

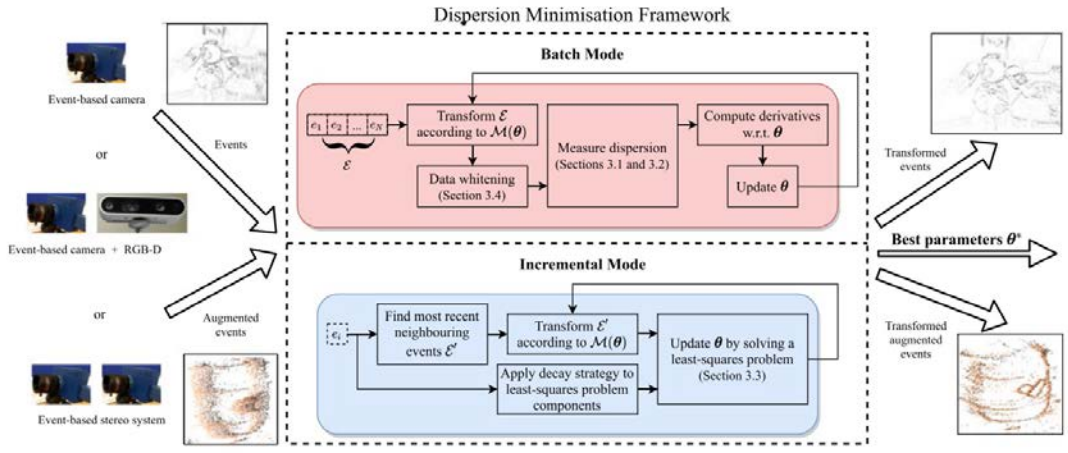


Figure 2.5: *Dispersion minimization framework.* Data from Fig. 1 of [113].

propose a branch-and-bound strategy to limit the search space of the homography transformation (8 DOFs) for a better convergence, although it needs to provide the initial search space. Stoffregen et al. [152] propose the expectation-maximization algorithm to estimate multiple clusters with 2 DOFs (i.e., *motion segmentation*) using CMax, however, the convergence heavily depends on the initial conditions (especially for the maximization step). As the complexity (e.g., DOF of the motion parameter θ) increases, the problem becomes more difficult to solve (i.e., converge), which tends to need limiting the search space, relying on good initial conditions, or choosing the optimization algorithm carefully.

Similarly to CMax, Nunes and Demiris [112,113] formulate the event alignment problem via dispersion minimization (DMin) (Fig. 2.5). The dispersion (e.g., entropy) measures pairwise distances between feature points (e.g., warped events, which can be either in two- or three-dimensional coordinates). By doing so, they demonstrate the 6-DOF estimation capability if the GT depth information is externally provided. Notice that the essential difference between DMin and CMax lies in the calculation of the event alignment: DMin requires the pairwise distance calculation that takes $O(N_e^2)$ without any approximation (N_e denotes the number of events), while CMax successfully reduces the complexity to $O(N_e + N_p)$ by using IWEs as an intermediate representation to measure the alignment.

In contrast to the optimization-based methods, learning-based ego-motion estimation from events is relatively unexplored. Gehrig et al. [51] propose a spiking neural network (SNN) model that estimates the angular velocity (i.e., 3 DOFs) in a supervised manner (Fig. 2.6). However, it requires precise ground truth signals, which they address by using simulation datasets. Another supervised learning for the 6-DOF ego-motion is proposed in [111], where they apply a Long-Short Term Memory (LSTM) network.

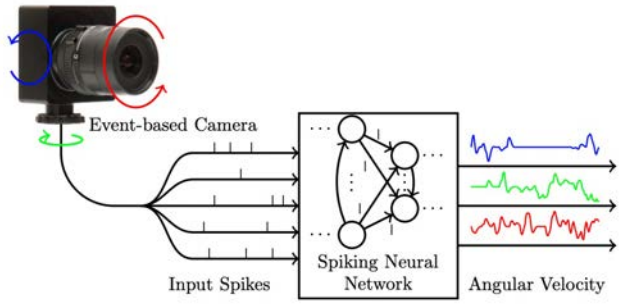


Figure 2.6: *Supervised learning of angular velocity using Spiking Neural Network.* Data from Fig. 1 of [51].

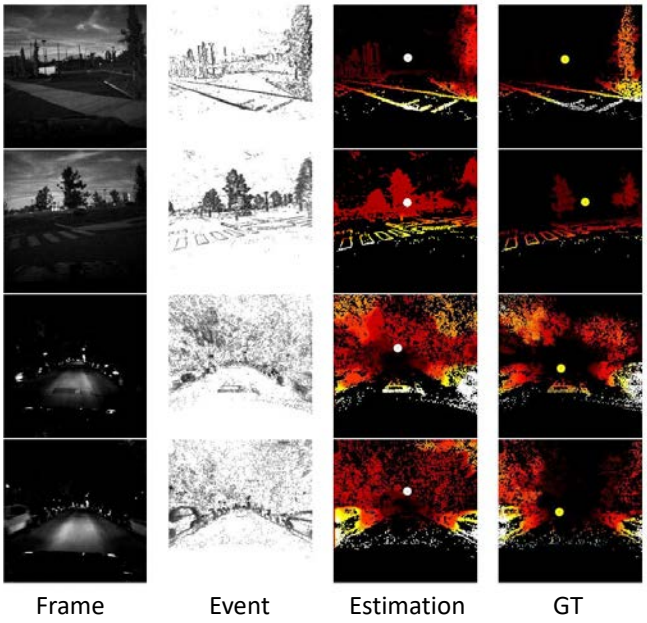


Figure 2.7: *Unsupervised learning of depth and ego-motion.* Data from Fig. 6 of [180].

Learning-based approaches have also tackled the "ideal" case of the problem, which is the simultaneous estimation of the pose with the scene depth. Ye et al. [170] and Zhu et al. [180] propose a joint estimation based on unsupervised learning. They both convert event stream into image- or tensor-representation as the input to the DNN. Then, Ye et al. [170] use the photometric loss by warping the input image representation with the estimated ego-motion and depth. Zhu et al. [180] propose the average timestamp loss (the sum of squares of the average timestamp at each pixel) as a proxy function for the contrast functions in CMax (Fig. 2.7).

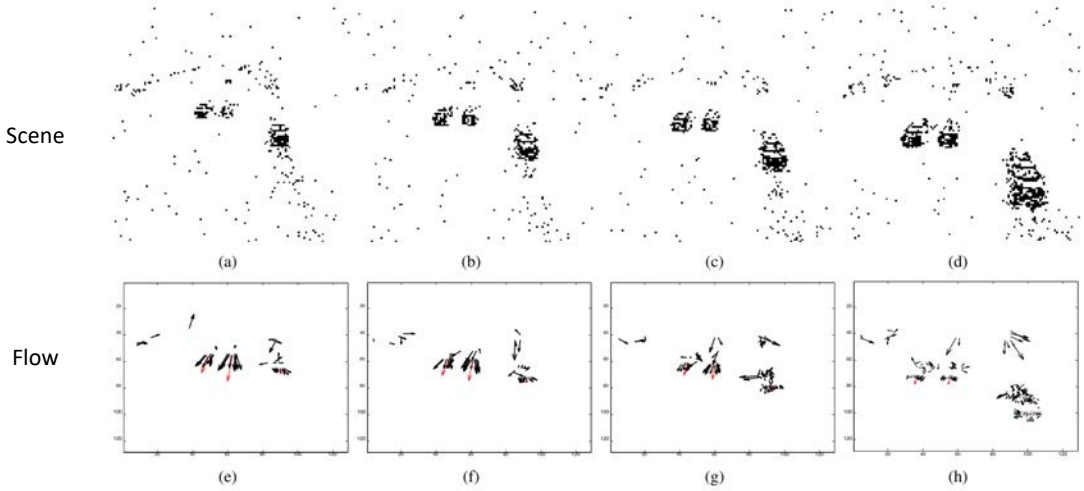


Figure 2.8: *An example of optimization-based optical flow estimation.* Data from Fig. 10 of [10].

However, they report the overfit of the network when the original contrast functions are used, which is one of the problems that we challenge in this work (Chapter 4).

2.3 Motion Estimation from Events: Optical Flow Estimation

Optical flow is pixel displacements between two images within a short time interval such that the displacements do not change during the interval (i.e., linear). It has generally higher DOFs (e.g., $2N_p$) than that of the pose. Indeed, the previous section can be seen as the particularization of optical flow estimation when the scene is static. Although it can represent complex motion even with independently moving objects, event-based optical flow is a challenging task due to its high complexity.

Prior optimization-based work has proposed adaptations of frame-based approaches (block matching [96], Lucas-Kanade [11]), filter-banks [114,16], spatio-temporal plane-fitting [10,1], and time surface matching [109]. For example, Benosman et al. [10] formulate the flow as the inverse of spatial derivatives of co-occurring event timestamps (Fig. 2.8). Brosch et al. [16] propose the spatio-temporal filter-bank approach that is inspired by the motion detection mechanism from biology (Fig. 2.9). Notice that these methods estimate sparse optical flow (optical flow is estimated where at least one event exists). Bardow et al. [6] propose a simultaneous estimation of dense optical flow (flow over the entire image plane) and image intensity (Fig. 2.10). These relatively early works have been tested on rather simple scenes, as opposed to the more complex real-world scenes and publicly-available benchmarks.

Recently, there have been more learning-based approaches [179,180,56,34,87,65], largely inspired

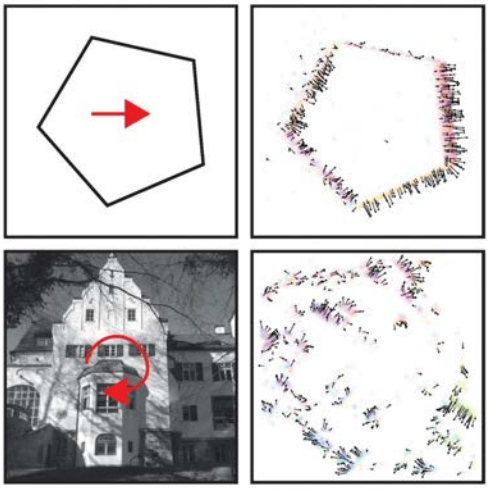


Figure 2.9: *An example of optimization-based optical flow estimation.* Data from Fig. 8 of[16].

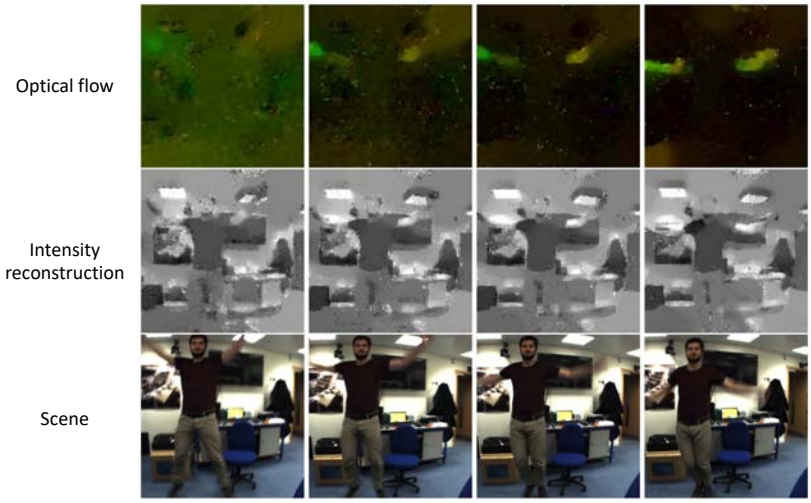


Figure 2.10: *An example of optimization-based optical flow estimation.*Optical flow is simultaneously solved with image reconstruction. Data from Fig. 6 of[6].

by frame-based optical flow architectures[134,160]. Non-spiking–based approaches need to additionally adapt the input signal, converting the events into a tensor representation (event frames, time surfaces, voxel grids, etc.[52,18]). These learning-based methods can be classified

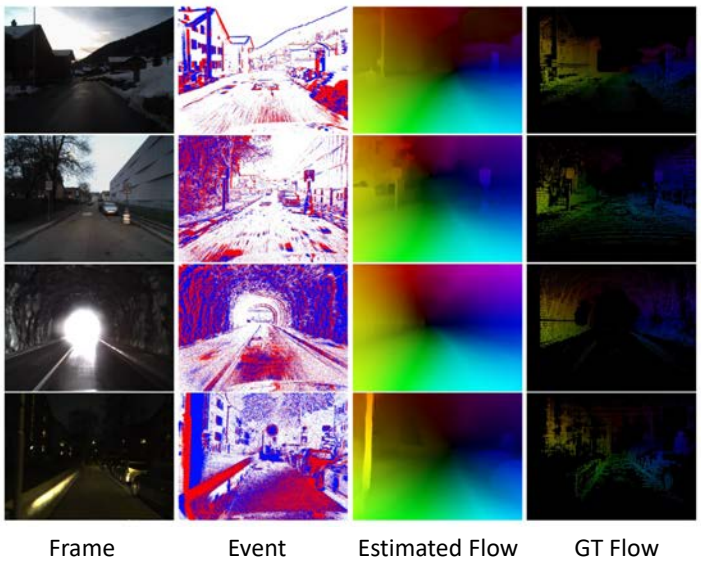


Figure 2.11: *An example of supervised optical flow estimation.* Data from Fig. 4 of Gehrig et al.[56]. Supervised learning relies on the ground truth (GT) flow. Notice the sparsity of the GT flow, which we will discuss in Chapter 4.

into supervised, semi-supervised, or unsupervised. In terms of architecture, the three most common ones are U-Net[179], FireNet[180] and RAFT[56].

Supervised methods train DNNs in simulation and/or real-data[56,155,52]. For example, Gehrig et al.[56] adapted the RAFT architecture[160], which is one of the state-of-the-art architectures in frame-based optical flow estimation (Fig. 2.11). The supervised learning requires accurate ground truth flow that matches the space-time resolution of event cameras. While this is no problem in simulation, it incurs a performance gap when trained models are used to predict flow on real data[155]. Besides, real-world datasets have issues in providing accurate ground truth flow. Although it will not be visible during the evaluation with the ground truth, supervised-learning methods might not be able to learn pixels without GT, such as independently moving objects and out of LiDAR's field of view (see the last column of Fig. 2.11). This also indicates that we need proxy metrics for the flow evaluation that do not depend on the GT, as we discuss in Chapter 4.

Semi-supervised methods use the grayscale images from a co-located camera (e.g., DAVIS[158]) as a supervisory signal: images are warped using the flow predicted by the DNN and their photometric consistency is used as loss function[179,87,34]. While such a supervisory signal is easier to obtain than real-world ground truth flow, it may suffer from the limitations of frame-based cameras (e.g., low dynamic range and motion blur), consequently affecting the

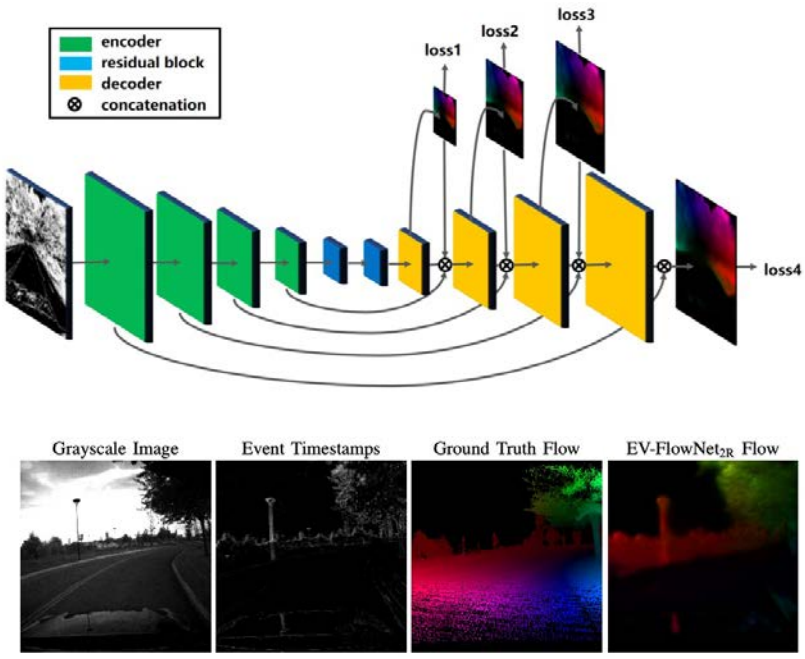


Figure 2.12: *An example of self-supervised optical flow estimation.* Data from Fig. 3 and Fig. 4 of Zhu et al.[178].

trained DNNs. These approaches were pioneered by EV-FlowNet[179], which is based on the U-Net architecture (Fig. 2.12).

Unsupervised methods rely solely on event data. Their loss function consists of an event alignment error using the flow predicted by the DNN[180,118,65,170]. Zhu et al.[180] extend EV-FlowNet[179] to the unsupervised setting using the average timestamp loss function. This approach has been used and improved in[118,65]. Tian et al.[161] use the same loss function with a transformer-based network architecture. Paredes-Vallés et al.[118] also propose FireFlowNet, a lightweight DNN producing competitive results, and jointly solve the image reconstruction and flow estimation (Fig. 2.13). Although these work produce competitive results with semi-supervised methods, the loss functions that have been used (average timestamp[180] and normalized average timestamp[65]) are unstable and difficult to interpret as the contrast functions.

2.4 Motion Estimation from Events and Frames

Thanks to cameras that output both events and frames with co-located pixels (e.g., DAVIS[158]), it has been possible to utilize the information of both data modalities to estimate motion from a single camera. In the literature, the following problem settings have been tackled:

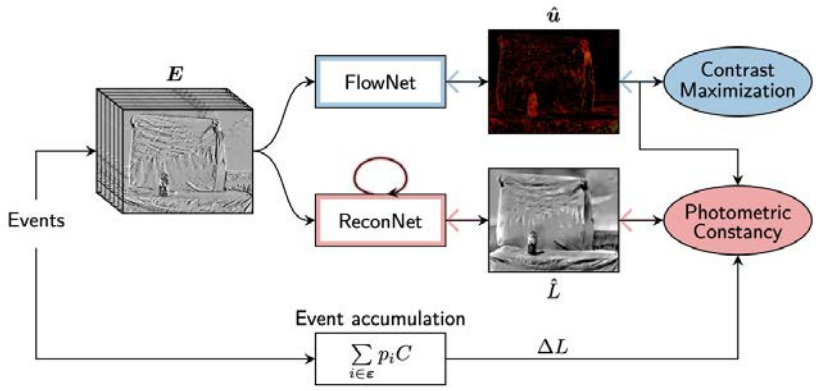


Figure 2.13: *An example of unsupervised optical flow estimation.* Data from Fig. 1 of Paredes-Vallés et al.[118].

(*i*) feature-based tracking[159,85,53,54], (*ii*) single-object–based tracking[95,171,166], and (*iii*) rigid-body structure from motion[17,71]. The resulting problems differ not only in increasing complexity but also in purpose: for example, object-based tracking is concerned with determining bounding boxes, while feature-based tracking aims at a sub-pixel precision marking of keypoints to enable SLAM applications.

Gehrig et al. propose Event-based Lucas-Kanade tracking that utilizes the linearized event generation model (LEGM)[53,54]. It extracts features from frames[98,5] and subsequently tracks them asynchronously using events (Fig. 2.14). The LEGM states that, the brightness increment (the per-pixel sum of the event polarities) is caused by the gradients of frames moving with image velocity. It has been extended toward more complex (rigid-body) motion in[17,71]. However, estimating further complex motion (e.g., optical flow) using the LEGM is yet to be explored, which we challenge in Chapter 6.

2.5 Conclusion

In this chapter, we review the various motion estimation problems using an event camera. Ego-motion estimation problems assume the static scene, where the "ideal" parameterization becomes $6 + N_p$. Simplified problem settings of the low-DOF motion estimation, such as feature flow and rotational motion, can be estimated using the event alignment methods such as the Contrast Maximization framework (Sec. 2.2). Optical flow estimation can handle independently moving objects in the scene with larger complexity. Various optical flow estimation methods have been proposed, from optimization-based ones for the sparse flow to supervised-learning ones for the dense flow (Sec. 2.3). Finally, motion estimation methods combining events and frames have been reviewed (Sec. 2.4).

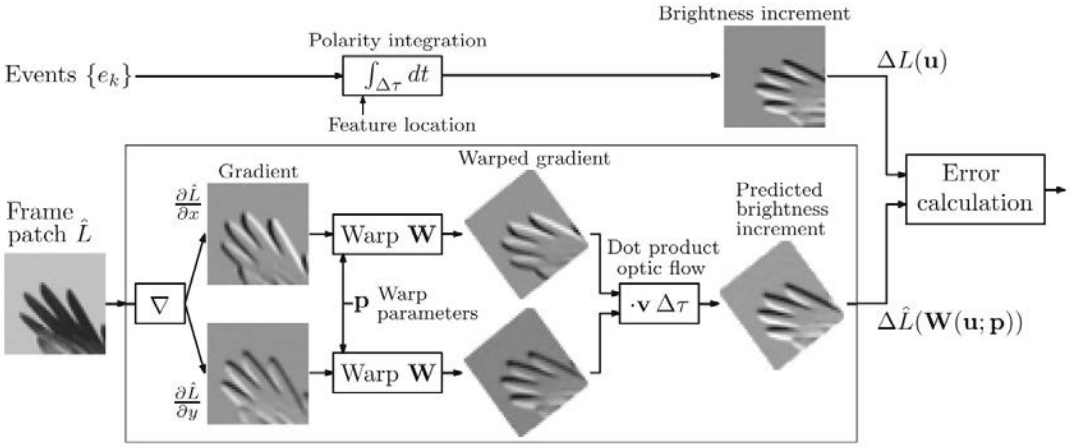


Figure 2.14: *Event-based Lucas-Kanade tracking.* Data from Fig. 5 of [54].

3 Low-DoF Motion Estimation

3.1 Introduction

Event cameras [30,156,41] offer potential advantages over standard cameras to tackle difficult scenarios (high speed, high dynamic range, low power). However, new algorithms are needed to deal with the unconventional type of data they produce (per-pixel asynchronous brightness changes, called *events*) and unlock their advantages [43]. Contrast maximization (CMax) is an event processing framework that provides state-of-the-art results on several tasks, such as rotational motion estimation [47,80], feature flow estimation and tracking [176,177,140,154,29], egomotion estimation [46,45,121], 3D reconstruction [46,128], optical flow estimation [180,120,65,146], motion segmentation [99,152,174,117,97], guided filtering [36], and image reconstruction [173]. The main idea of CMax and similar event alignment frameworks [113,63] is to find the motion and/or scene parameters that align corresponding events (i.e., events that are triggered by the same scene edge), thus achieving motion compensation. The framework simultaneously estimates the motion parameters and the correspondences between events (data association). However, in some cases CMax optimization converges to an undesired solution where events accumulate into too few pixels, a phenomenon called *event collapse* (Fig. 3.1). Since CMax is at the heart of many state-of-the-art event-based motion estimation methods, it is important to understand the above limitation and propose ways to overcome it. Prior works have largely ignored the issue or proposed workarounds without analyzing the phenomenon in detail. A more thorough discussion of the phenomenon is overdue, which is the goal of this chapter.

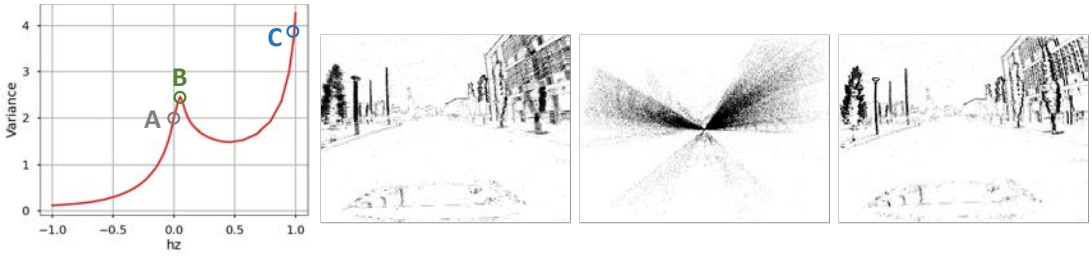


Figure 3.1: *Event Collapse*: Left: Landscape of the image variance loss as a function of the warp parameter h_z. Right: The IWEs at the different h_z marked in the landspace: A. Original events (identity warp), accumulated over a small Δt (polarity is not used). C. Image of warped events (IWE) showing event collapse due to maximization of the objective function. B. Desired IWE solution using our proposed regularizer: sharper than (A) while avoiding event collapse (C).

Contrarily to the expectation that event collapse occurs when the event transformation becomes sufficiently complex [180,113], we show that it may occur even in the simplest case of one degree-of-freedom (DOF) motion. Drawing inspiration from differential geometry and electrostatics we propose principled metrics to quantify event collapse and discourage it by incorporating penalty terms in the event alignment objective function. While event collapse depends on many factors, our strategy aims at modifying the objective's landscape to improve the well-posedness of the problem and be able to use well-known, standard optimization algorithms.

In summary, the contributions of this chapter are:

1. A study of the event collapse phenomenon in regards to event warping and objective functions (Secs. 3.3.1 and 3.6).

2. Three principled metrics of event collapse (one based on flow divergence and two based on area-element deformations) and their use as regularizers to mitigate the above-mentioned phenomenon (Secs. 3.2.3, 3.4 and 3.5).

3. Experiments on publicly available datasets that demonstrate, in comparison with other strategies, the effectiveness of the proposed regularizers (Sec. 3.6).

To the best of our knowledge, this is the first work that focuses on the paramount phenomenon of event collapse, which may arise in state-of-the-art event-alignment methods. Our experiments show that the proposed metrics mitigate event collapse while they do not harm well-posed warps or trade-off the algorithm runtime.

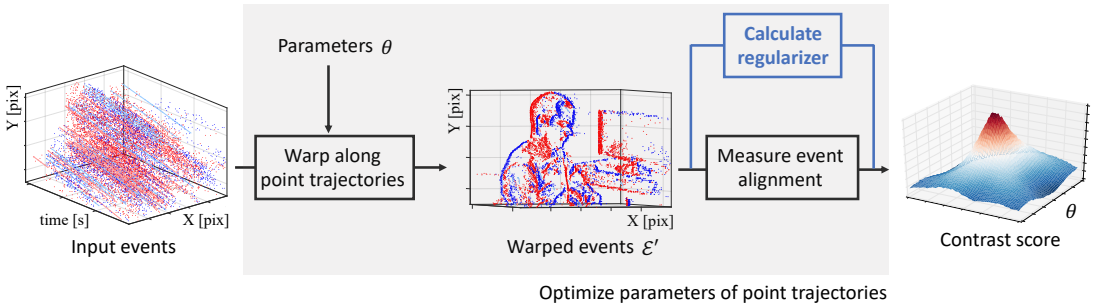


Figure 3.2: Proposed modification of the Contrast Maximization (CMax) framework in [46,45] to also account for the degree of regularity (collapsing behavior) of the warp. Events are colored in red/blue according to their polarity.

3.2 Contrast Maximization

In this section, first, we revise how event cameras work (Sec. 3.2.1) and the CMax framework (Sec. 3.2.2). Then, Sec. 3.3.1 builds our intuition on event collapse by analyzing a simple example. Section 3.4 presents our proposed metrics for event collapse, based on 1-DOF and 2-DOF warps. Section 3.5 specifies them for higher DOFs, and Sec. 3.2.3 presents the regularized objective function.

3.2.1 How Event Cameras Work

Event cameras, such as the Dynamic Vision Sensor (DVS)[91,156,41], are bio-inspired sensors that capture pixel-wise *intensity changes*, called events, instead of intensity images. An event $e_k \doteq (\mathbf{x}_k, t_k, p_k)$ is triggered as soon as the logarithmic intensity L at a pixel exceeds a contrast sensitivity $C > 0$,

$$L(\mathbf{x}_k, t_k) - L(\mathbf{x}_k, t_k - \Delta t_k) = p_k\, C, \tag{3.1}$$

where $\mathbf{x}_k \doteq (x_k, y_k)^\top$, t_k (with μs resolution) and polarity $p_k \in \{+1, -1\}$ are the spatio-temporal coordinates and sign of the intensity change, respectively, and $t_k - \Delta t_k$ is the time of the previous event at the same pixel $\mathbf{x}_k$. Hence, each pixel has its own sampling rate, which depends on the visual input.

3.2.2 Mathematical Description of the CMax Framework

The CMax framework[46] transforms events in a set $\mathcal{E} = \{e_k\}_{k=1}^{N_e}$ geometrically

$$e_k \doteq (\mathbf{x}_k, t_k, p_k) \quad \overset{\mathbf{W}}{\mapsto} \quad e'_k \doteq (\mathbf{x}'_k, t_{\text{ref}}, p_k), \tag{3.2}$$

according to a motion model $\mathbf{W}$, producing a set of warped events $\mathcal{E}' = \{e'_k\}_{k=1}^{N_e}$. The warp $\mathbf{x}'_k = \mathbf{W}(\mathbf{x}_k, t_k; \theta)$ transports each event along the point trajectory that passes through it (Fig. 3.2, left), until t_{ref} is reached. The point trajectories are parametrized by θ, which contains the motion and/or scene unknowns. Then, an objective function[45,154] measures the alignment of the warped events $\mathcal{E}'$. Many objective functions are given in terms of the count of events along the point trajectories, which is called the image of warped events (IWE):

$$I(\mathbf{x}; \theta) \doteq \sum_{k=1}^{N_e} b_k \, \delta(\mathbf{x} - \mathbf{x}'_k(\theta)). \tag{3.3}$$

Each IWE pixel $\mathbf{x}$ sums the values of the warped events $\mathbf{x}'_k$ that fall within it: $b_k = p_k$ if polarity is used or $b_k = 1$ if polarity is not used. The Dirac delta δ is in practice replaced by a smooth approximation[110], such as a Gaussian, $\delta(\mathbf{x} - \mu) \approx \mathcal{N}(\mathbf{x}; \mu, \varepsilon^2)$ with $\varepsilon = 1$ pixel. A popular objective function $G(\theta)$ is the visual contrast of the IWE (3.3), given by the variance

$$G(\theta) \equiv \text{Var}\big(I(\mathbf{x}; \theta)\big) \doteq \frac{1}{|\Omega|} \int_\Omega (I(\mathbf{x}; \theta) - \mu_I)^2 d\mathbf{x}, \tag{3.4}$$

with mean $\mu_I \doteq \frac{1}{|\Omega|} \int_\Omega I(\mathbf{x}; \theta) d\mathbf{x}$ and image domain Ω. Hence, the alignment of the transformed events $\mathcal{E}'$ (i.e., the candidate "corresponding events", triggered by the same scene edge) is measured by the strength of the edges of the IWE. Finally, an optimization algorithm iterates the above steps until the best parameters are found:

$$\theta^* = \arg\max_\theta G(\theta). \tag{3.5}$$

3.2.3 Augmented Objective Function

We propose to augment previous objective functions (e.g., (3.5)) with penalties obtained from the above developed metrics for event collapse:

$$\theta^* = \operatorname{argmin}_\theta J(\theta) = \operatorname{argmin}_\theta \left(-G(\theta) + \lambda \mathcal{R}(\theta)\right). \tag{3.6}$$

We may interpret $G(\theta)$ (e.g., contrast or focus score[45]) as the data fidelity term and $\mathcal{R}(\theta)$ as the regularizer, or, in Bayesian terms, the likelihood and the prior, respectively.

3.3 Event Collapse

3.3.1 Simplest Example of Event Collapse: 1 DOF

To analyze event collapse in the simplest case, let us consider an approximation to a translational motion of the camera along its optical axis Z (1-DOF warp). In theory, translational motions require also the knowledge of the scene depth. Here, inspired by the 4-DOF in-plane warp in[99] that approximates a 6-DOF camera motion, we consider a simplified warp that does not require knowledge of the scene depth. In terms of data, let us consider events from one of the driving sequences of the standard MVSEC dataset[178] (Fig. 3.1).

For further simplicity, let us normalize the timestamps of $\mathcal{E}$ to the unit interval $t \in [t_1, t_{N_e}] \mapsto \tilde{t} \in [0, 1]$, and assume a coordinate frame at the center of the image plane, then the warp $\mathbf{W}$ is given by

$$\mathbf{x}'_k = (1 - \tilde{t}_k h_z)\, \mathbf{x}_k, \tag{3.7}$$

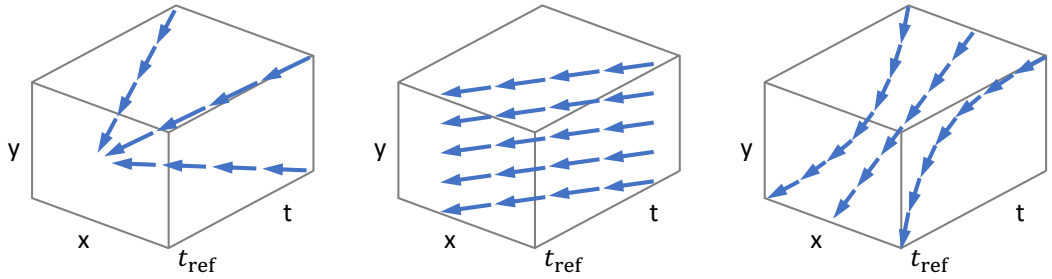
where $\theta \equiv h_z$. Hence, events are transformed along the radial direction from the image center, acting as a virtual focus of expansion (FOE) (cf. the true FOE is given by the data). Letting the scaling factor in (3.7) be $s_k \doteq 1 - \tilde{t}_k h_z$, we observe the following: (*i*) s_k cannot be negative since it would imply that at least one event has flipped the side on which it lies with respect to the image center. (*ii*) if $s_k > 1$ the warped event gets away from the image center ("expansion" or "zoom-in"). (*iii*) if $s_k \in [0, 1)$ the warped event gets closer to the image center ("contraction" or "zoom-out"). The equivalent conditions in terms of h_z are: (*i*) $h_z < 1$, (*ii*) $h_z < 0$ is an expansion, (*iii*) $0 < h_z < 1$ is a contraction.

Intuitively, event collapse occurs if the contraction is large ($0 < s_k \ll 1$) (see Fig. 3.1C, Fig. 3.3a). This phenomenon is not specific of the image variance; other objective functions lead to the same result. As we see, the objective function has a local maximum at the desired motion parameters (Fig. 3.1B). The optimization over the entire parameter space converges to a global optimum that explains the event collapse.

3.3.2 Discussion

The above example shows that event collapse is enabled (or disabled) by the type of warp. If the warp does not enable event collapse (contraction or accumulation of flow vectors cannot happen due to the geometric properties of the warp), as in the case of feature flow (2 DOFs)[176,153] (Fig. 3.3b) or rotational motion flow (3 DOFs)[47,93] (Fig. 3.3c), then the optimization problem is well posed and multiple objective functions can be designed to achieve event alignment[45,154]. However, the disadvantage is that the type of warps that satisfy this condition may not be rich enough to describe complex scene motions.

On the other hand, if the warp allows for event collapse, more complex scenarios can be described by such a broader class of motion hypotheses, but the optimization framework



(a) Zoom in/out warp from image center (1 DOF).

(b) Constant image velocity warp (2 DOF).

(c) Rotational warp around X axis (3 DOF).

Figure 3.3: *Point trajectories* (streamlines) defined on $x - y - t$ image space by various warps.

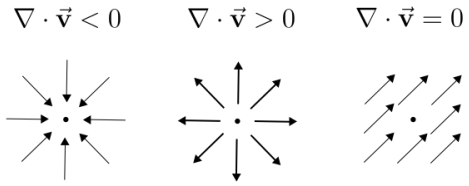


Figure 3.4: *Divergence* of different vector fields, $\nabla \cdot \mathbf{v} = \partial_x \mathbf{v}_x + \partial_y \mathbf{v}_y$. From left to right: contraction ("sink", leading to event collapse), expansion ("source"), and incompressible fields. Image adapted from khanacademy.org

designed for non-event-collapsing scenarios (where the local maximum is assumed to be the global maximum) may not hold anymore. Optimizing the objective function may lead to an undesired solution with a larger value than the desired one. This depends on multiple elements: the landscape of the objective function (which depends on the data, the warp parametrization and the shape of the objective function), and the initialization and search strategy of the optimization algorithm used to explore such a landscape. The challenge in this situation is to overcome the issue of multiple local maxima and make the problem better posed. Our approach consists of characterizing event collapse via novel metrics and including them in the objective function as weak constraints (penalties) to yield a better landscape.

3.4 Proposed Regularizers

3.4.1 Divergence of the Event Transformation Flow

Inspired by physics, we may think of the flow vectors given by the event transformation $\mathcal{E} \mapsto \mathcal{E}'$ as an electrostatic field, whose sources and *sinks* correspond to the location of electric charges (Fig. 3.4). Sources and sinks are mathematically described by the divergence operator $\nabla \cdot$. Therefore, the divergence of the flow field is a natural choice to characterize event

collapse.

The warp $\mathbf{W}$ is defined over the space-time coordinates of the events, hence its time derivative defines a flow field over space-time:

$$\mathbf{f} \doteq \frac{\partial \mathbf{W}(\mathbf{x}, t; \theta)}{\partial t}. \tag{3.8}$$

For the warp in (3.7), we obtain $\mathbf{f} = -h_z \mathbf{x}$, which gives $\nabla \cdot \mathbf{f} = -h_z \nabla \cdot \mathbf{x} = -2h_z$. Hence, (3.7) defines a constant divergence flow, and imposing a penalty on the degree of concentration of the flow field accounts to directly penalizing the value of the parameter h_z.

Computing the divergence at each event gives the set

$$\mathcal{D}(\mathcal{E}, \theta) \doteq \{\nabla \cdot \mathbf{f}_k\}_{k=1}^{N_e}, \tag{3.9}$$

from which we can compute statistical scores (mean, median, min, etc.):

$$R_D(\mathcal{E}, \theta) \doteq \frac{1}{N_e} \sum_{k=1}^{N_e} \nabla \cdot \mathbf{f}_k. \qquad \text{(mean)} \tag{3.10}$$

To have a 2D visual representation ("feature map") of collapse, we build an image (like the IWE) by taking some statistic of the values $\nabla \cdot \mathbf{f}_k$ that warp to each pixel, such as the "average divergence per pixel":

$$\mathrm{DIWE}(\mathbf{x}; \mathcal{E}, \theta) \doteq \frac{1}{N_e(\mathbf{x})} \sum_k (\nabla \cdot \mathbf{f}_k)\, \delta(\mathbf{x} - \mathbf{x}'_k), \tag{3.11}$$

where $N_e(\mathbf{x}) \doteq \sum_k \delta(\mathbf{x} - \mathbf{x}'_k)$ is the number of warped events at pixel $\mathbf{x}$ (the IWE). Then we aggregate further into a score, such as the mean:

$$R_{\mathrm{DIWE}}(\mathcal{E}, \theta) \doteq \frac{1}{|\Omega|} \int_\Omega \mathrm{DIWE}(\mathbf{x}; \mathcal{E}, \theta) d\mathbf{x}. \tag{3.12}$$

In practice we focus on the collapsing part by computing a trimmed mean: the mean of the DIWE pixels smaller than a margin α (-0.2 in the experiments). Such a margin does not penalize small, admissible deformations.

3.4.2 Area-based Deformation of the Event Transformation

Besides vector calculus, we may also use tools from differential geometry to characterize event collapse. Building on[46], the point trajectories define the streamlines of the transformation

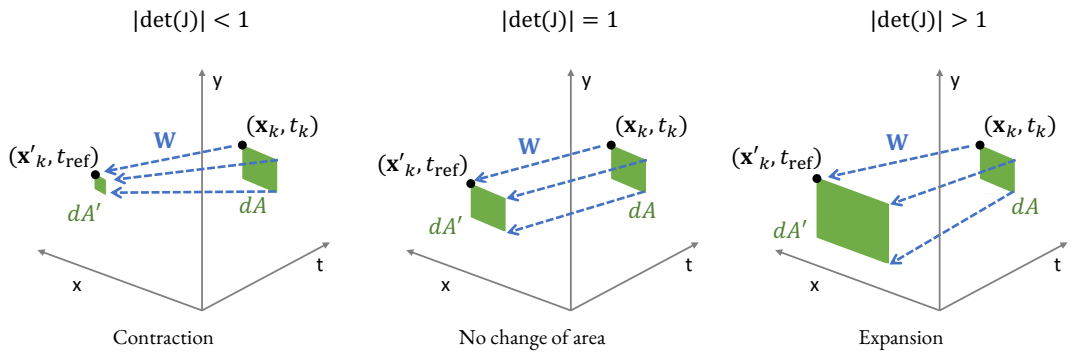


Figure 3.5: *Area deformation* of various warps. An area of dA pix^2 at $(\mathbf{x}_k, t_k)$ and is warped to t_{ref}, giving an area $dA' = |\det(\mathtt{J}_k)|dA$ pix^2 at $(\mathbf{x}'_k, t_{\text{ref}})$, where $\mathtt{J}_k \equiv \mathtt{J}(e_k) \equiv \mathtt{J}(\mathbf{x}_k, t_k; \theta)$ (see (3.13)). From left to right, increasing area amplification factor $|\det(\mathtt{J})| \in [0, \infty)$.

flow, and we may measure how they concentrate or disperse based on how the area element deforms along them. That is, we consider a small area element $dA = dxdy$ attached to each point along the trajectory and measure how much it deforms when transported to the reference time: $dA' = |\det(\mathtt{J})|\, dA$, with Jacobian

$$\mathtt{J}(\mathbf{x}, t; \theta) \doteq \frac{\partial \mathbf{W}(\mathbf{x}, t; \theta)}{\partial \mathbf{x}} \tag{3.13}$$

(see Fig. 3.5). The determinant of the Jacobian is the amplification factor: $|\det(\mathtt{J})| > 1$ if the area expands, and $|\det(\mathtt{J})| < 1$ if the area shrinks.

For the warp in (3.7), we have the Jacobian $\mathtt{J} = (1 - \tilde{t}h_z)\mathtt{Id}$, and so $\det(\mathtt{J}) = (1 - \tilde{t}h_z)^2$. Interestingly, the area deformation around event e_k, $\mathtt{J}(e_k) \equiv \mathtt{J}(\mathbf{x}_k, t_k; \theta)$, is directly related to the scaling factor s_k: $\det(\mathtt{J}(e_k)) = s_k^2$.

Computing the amplification factors at each event gives the set

$$\mathcal{A}(\mathcal{E}, \theta) \doteq \left\{ |\det(\mathtt{J}(e_k))| \right\}_{k=1}^{N_e}, \tag{3.14}$$

from which we can compute statistical scores. For example,

$$R_A(\mathcal{E}, \theta) \doteq \frac{1}{N_e} \sum_{k=1}^{N_e} |\det(\mathtt{J}(e_k))| \qquad \text{(mean)} \tag{3.15}$$

gives an average score: $R_A > 1$ for expansion, and $R_A < 1$ for contraction.

We build a deformation map (or image of warped areas (IWA)) by taking some statistic of the values $|\det(\mathtt{J}(e_k))|$ that warp to each pixel, such as the "average amplification per pixel":

$$\mathrm{IWA}(\mathbf{x}) \doteq 1 + \frac{1}{N_e(\mathbf{x})} \sum_{k=1}^{N_e} \big(|\det(\mathrm{J}(e_k))| - 1\big)\, \delta(\mathbf{x} - \mathbf{x}'_k). \tag{3.16}$$

This assumes that if no events warp to a pixel $\mathbf{x}_p$, then $N_e(\mathbf{x}_p) = 0$, and there is no deformation ($\mathrm{IWA}(\mathbf{x}_p) = 1$). Then, we summarize the deformation map into a score, such as the mean:

$$R_{\mathrm{IWA}}(\mathcal{E}, \theta) \doteq \frac{1}{|\Omega|} \int_{\Omega} \mathrm{IWA}(\mathbf{x}; \mathcal{E}, \theta) d\mathbf{x}. \tag{3.17}$$

To concentrate on the collapsing part, we compute a trimmed mean: the mean of the IWA pixels smaller than a margin α (0.8 in the experiments). The margin approves small, admissible deformations.

3.4.3 Rate of Change of Area Deformation

The complexity of the previous two regularizers is $O(N_e + N_p)$ because (3.9) and (3.14) depend linearly on the number of events N_e, and the resulting average images ((3.11) and (3.16)) have N_p pixels. This extra complexity makes the whole CMax pipeline more than twice slower than the original (unregularized) CMax framework, whose complexity is also $O(N_e + N_p)$[46]. Not only the computational complexity is a burden, but also the fact that (3.14) are measured relative to a single reference time. For example, $\mathcal{A}(\mathcal{E}, \theta)$ (3.14) increases as t_k increases, since it measures the area deformation *from* t_k *to* $t_{ref} = t_1$. This scaling problem is undesirable because (*i*) events far from t_{ref} contribute more to $\mathcal{R}$ than events closer to t_{ref}, and (*ii*) this effect could be amplified depending on the temporal distribution of the events.

Intuitively, motion fields are well-posed or not (i.e., collapse-enabled) by design of the problem, regardless of the event data. Hence, an ideal regularizer should not depend on the events, but solely on the warp parameters (Fig. 3.6, blue line). The main idea of the third proposed regularizer is to aggregate differential deformations rather than relative ones. Figure 3.7 shows the geometric interpretation: $\mathcal{R}$ is obtained as the integral of the rate-of-change of the area element deformation along the space-time point trajectories $(\mathbf{x}(t), t)$ defined by the motion.

Again, for simplicity, consider the 1-DOF motion estimation (3.7). Assuming an area element attached to each point of the motion trajectory $\gamma(t) = (\mathbf{x}(t), t)$ (Fig. 3.7), the change of area (i.e., area deformation) from t to $t + \Delta t$ is given by:

$$|\mathrm{J}_{t,t+\Delta t}| \doteq \left|\det\left(\frac{d\mathbf{x}(t+\Delta t)}{d\mathbf{x}(t)}\right)\right| = \left(\frac{1 - t h_z}{1 - (t+\Delta t) h_z}\right)^2. \tag{3.18}$$

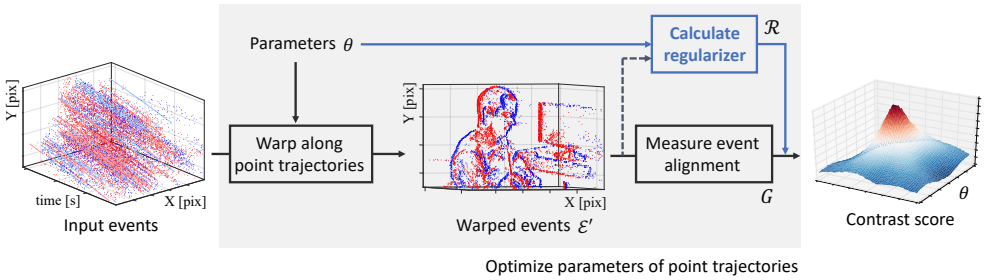


Figure 3.6: *Overview of the efficient regularization.* The proposed regularizer (blue line) solely relies on motion parameters θ, while previous approaches (dashed line) are built from warped events (see also: Fig. 3.2)[145].

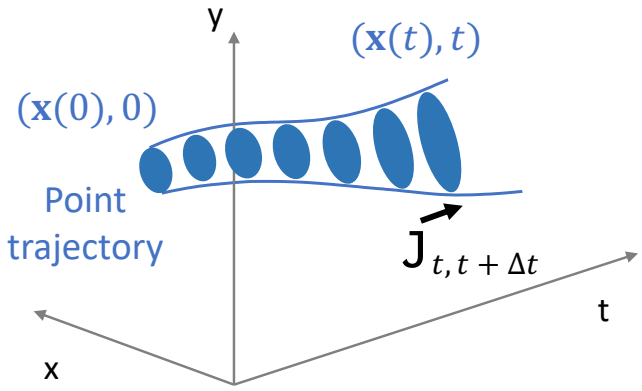


Figure 3.7: *Rate of change of area deformation.* The warp $\mathbf{W}$ defines point trajectories $\gamma(t) = (\mathbf{x}(t), t)$ in the space-time image domain. We define the regularizer $\mathcal{R}$ based on differential area deformation along $\gamma(t)$. The rate of change of area is given by the derivative of the Jacobian $\mathrm{J}_{t,t+\Delta t}$.

The Taylor series expansion of (3.18) at $\Delta t = 0$ is

$$|\mathrm{J}_{t,t+\Delta t}| = 1 + \left.\frac{d|\mathrm{J}_{t,t+\Delta t}|}{d\Delta t}\right|_{\Delta t=0} \Delta t + \cdots \tag{3.19}$$

Since the first term is always 1 (i.e., is trivial), we focus on the second term, which conveys the meaning of "speed" of area deformation. The derivative of (3.18) at $\Delta t = 0$ conveys the *rate of change* or *differential amplification factor* of the area:

$$\left.\frac{d|\mathrm{J}_{t,t+\Delta t}|}{d\Delta t}\right|_{\Delta t=0} = \frac{2h_z}{1 - th_z}. \tag{3.20}$$

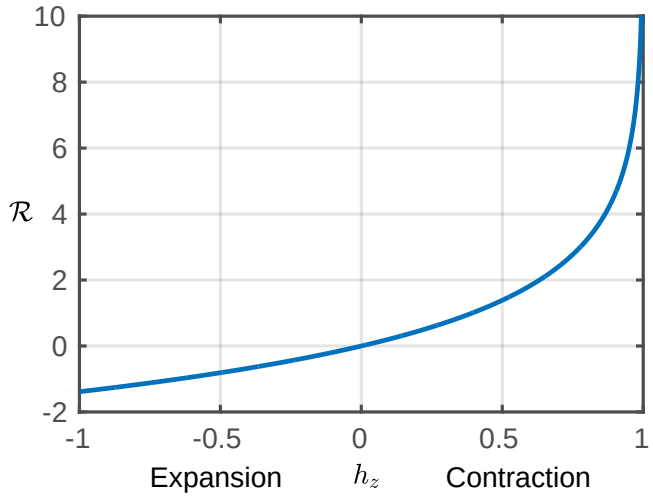


Figure 3.8: Regularizer $\mathcal{R}$ for the 1-DOF warp, (3.21).

Finally, the total rate of change of the deformation along the observation time window is

$$\mathcal{R} \doteq \int_0^1 \left.\frac{d|\mathrm{J}_{t,t+\Delta t}|}{d\Delta t}\right|_{\Delta t=0} dt \overset{(3.20)}{=} -2\log|1-h_z|. \tag{3.21}$$

The regularizer (3.21) is plotted in Fig. 3.8. It solely depends on $\theta \equiv h_z$ and has computational complexity $O(1)$. In addition, it is developed from geometric principles, and it is interpretable: $h_z = 0$ (identity warp) gives $\mathcal{R} = 0$; $h_z \in (0,1)$ (contraction; collapsing warp) gives large $\mathcal{R} > 0$; and $h_z < 0$ (expansion warp) gives $\mathcal{R} < 0$. Moreover, notice that $\mathcal{R}$ behaves like a barrier function, approaching infinity (i.e., large penalty) for values close to $h_z = 1$ (maximum allowed contraction before events flip side with respect to the image center).

3.5 Higher-DOF Warp Models

3.5.1 Feature Flow

Event-based feature tracking is often described by the warp $\mathbf{W}(\mathbf{x}, t; \theta) = \mathbf{x} + (t - t_{\text{ref}})\theta$, which assumes constant image velocity $\theta \equiv (v_x, v_y)^\top$ (2 DOFs) over short time intervals for all pixels. The 2-DOF translational motion (feature flow) in image space is a well-posed warp, since collapse never happens because the motion lines are parallel. As expected, the flow for this warp coincides with the image velocity, $\mathbf{f} = \theta$, which is independent of the space-time coordinates $(\mathbf{x}, t)$.

Divergence. The flow is incompressible ($\nabla \cdot \mathbf{f} = 0$): the streamlines given by the feature flow do not concentrate or disperse; they are parallel.

Area deformation. Regarding the area deformation, the Jacobian $\mathtt{J} = \partial(\mathbf{x} + (t - t_{\text{ref}})\theta)/\partial\mathbf{x} = \mathtt{Id}$ is the identity matrix. Hence $|\det(\mathtt{J})| = 1$, that is, translations on the image plane do not change the area of the pixels around a point.

Rate of change of area deformation. The rate of change of the deformation is $\mathcal{R} \equiv \int_0^1 \frac{d|\mathtt{J}|}{d\Delta t} = 0$.

In-plane translation warps, such as the above 2-DOF warp, are well-posed and serve as reference to design the regularizers that measure event collapse. It is sensible for well-designed regularizers to penalize warps whose characteristics deviate from those of the reference warp: zero divergence and unit area amplification factor.

3.5.2 Rotational Motion

As previous sections show, the proposed metrics designed for the zoom in/out warp produce the expected characterization of the 2-DOF feature flow (zero divergence and unit area amplification), which is a well-posed warp. Hence, if they were added as penalties into the objective function they would not modify the energy landscape. We now consider their influence on rotational motions, which are also well-posed warps. In particular, we consider the problem of estimating the angular velocity of a predominantly rotating event camera by means of CMax, which is a popular research topic[47,93,121,113,63]. Using calibrated and homogeneous coordinates, the warp is given by

$$\mathbf{x}^{b\prime} \sim \mathtt{R}(t\omega)\,\mathbf{x}^{b}, \tag{3.22}$$

where $\theta \equiv \omega = (\omega_1, \omega_2, \omega_3)^\top$ is the angular velocity, $t \in [0, \Delta t]$, and $\mathtt{R}$ is parametrized using exponential coordinates (Rodrigues rotation formula[7,108,48]).

Divergence. It is well known that the flow is $\mathbf{f} = B(\mathbf{x})\,\omega$, where $B(\mathbf{x})$ is the rotational part of the feature sensitivity matrix[27]. Hence

$$\nabla \cdot \mathbf{f} = 3(x\omega_2 - y\omega_1). \tag{3.23}$$

Area deformation. Letting $\mathbf{r}_3^\top$ be the third row of $\mathtt{R}$, and using (32)-(34) in[49],

$$\det(\mathtt{J}) = (\mathbf{r}_3^\top \mathbf{x}^b)^{-3}. \tag{3.24}$$

Rotations around the Z axis clearly present no deformation, regardless of the amount of rotation, and this is captured by the proposed metrics because: (*i*) the divergence is zero, thus the flow is incompressible, and (*ii*) $\det(\mathtt{J}) = 1$ since $\mathbf{r}_3 = (0, 0, 1)^\top$ and $\mathbf{x}^b = (x, y, 1)^\top$.

For other, arbitrary rotations, there are deformations, but these are mild if the rotation angle $\Delta t\|\omega\|$ is small.

Rate of change of area deformation. The incremental rotation between t and $t+\Delta t$ yields

$$\mathbf{x}^b(t+\Delta t) \sim \mathtt{R}(\omega\Delta t)\,\mathbf{x}^b(t). \tag{3.25}$$

Hence, the area element at $\mathbf{x}(t)$ deforms according to:

$$\det\left(\frac{d\mathbf{x}(t+\Delta t)}{d\mathbf{x}(t)}\right) = \left(\mathbf{r}_3^\top(\omega\Delta t)\mathbf{x}^b(t)\right)^{-3}, \tag{3.26}$$

where $\mathbf{r}_3^\top$ is the third row of $\mathtt{R}(\omega\Delta t)$ (see Sec. A.3). The derivative of (3.26) at $\Delta t = 0$ is given by:

$$\left.\frac{d|\mathtt{J}_{t,t+\Delta t}|}{d\Delta t}\right|_{\Delta t=0} = 3\mathbf{x}^{b\top}(t)\omega^\wedge\mathbf{e}_3. \tag{3.27}$$

Finally, the integral of (3.27) over the point trajectory (parametrized by the initial point $\mathbf{x}(0)$) is given by:

$$\begin{aligned}\mathcal{R}_{\mathbf{x}(0)} &= \int_0^1 \left.\frac{d|\mathtt{J}_{t,t+\Delta t}|}{d\Delta t}\right|_{\Delta t=0} dt \\ &= 3\omega_x\int_0^1 y(t)dt - 3\omega_y\int_0^1 x(t)dt.\end{aligned} \tag{3.28}$$

The integrals in (3.28) have units of absement. To obtain the regularizer $\mathcal{R}$, we threshold $\mathcal{R}_{\mathbf{x}(0)}$ at -0.2, which allows small amounts of natural deformation caused by rotation. Similar to (3.21), (3.28) does not depend on the events. However, in contrast to (3.21), (3.28) is spatially varying, providing an aggregated deformation map: it is smaller in the center of the image and larger (in absolute value) in the periphery. The computational complexity of $\mathcal{R}$ is $O(N_p)$, which can be further reduced if only a subset of the pixels is used.

Although 3-DOF rotations involve small deformations, their values (3.27) are considerably smaller than those of collapse-enabled warps like (3.7), and $\mathcal{R}$ does not affect the accuracy of the angular velocity estimation (as Sec. 3.6.3 will show). Also, pure rotations around the Z axis $\omega = (0, 0, \omega_z)^\top$ do not change the area, as expected, resulting in $\mathcal{R} = 0$.

3.5.3 Planar Motion

Planar motion is the term used to describe the motion of a ground robot that can translate and rotate freely on a flat ground. If such a robot is equipped with a camera pointing upwards or downwards, the resulting motion induced on the image plane, parallel to the ground plane,

is an isometry (Euclidean transformation). This motion model is a subset of the parametric ones in [46], and it has been used for CMax in [121,113]. For short time intervals, planar motion may be parametrized by 3 DOFs: linear velocity (2 DOFs) and angular velocity (1 DOF). As shown in Appendix A, the planar motion is a well-posed warp. The resulting motion curves on the image plane do not lead to event collapse.

3.5.4 Similarity Transformation

The 1-DOF zoom in/out warp in Sec. 3.3.1 is a particular case of the 4-DOF warp in [99,113], which is an in-plane approximation to the motion induced by a freely moving camera (6 DOFs). The scaling parameter h_z of the similarity transformation controls the amount of zoom in/out, i.e., the amount of contraction/expansion of the warp. Hence, we use (3.21) to penalize the amount of contraction. A mathematical justification is given in Appendix A.

3.6 Experiments

We evaluate our method on publicly-available datasets, whose details are described in Sec. 3.6.1. First, Sec. 3.6.2 shows that the proposed regularizers improves the objective function landscapes, reducing the undesired optima that enable collapse. For this purpose we use driving datasets (MVSEC [178], DSEC [55]). Next, Sec. 3.6.3 shows that the regularizers do not harm well-posed warps. To this end, we use the ECD dataset [105]. Then, Sec. 3.6.4 compare and discuss the runtimes among the proposed and existing approaches. Section 3.6.5 conducts a sensitivity analysis of the regularizers. Finally, Secs. 3.6.6 and 3.6.7 demonstrate applications of the proposed approaches.

3.6.1 Evaluation Datasets and Metrics

Datasets. The *MVSEC* dataset [178] is a widely-used dataset for various vision tasks, such as optical flow estimation [180,56,109,65,146]. Its sequences are recorded on a drone (indoors) or on a car (outdoors), and comprise events, grayscale frames and IMU data from a mDAVIS346 [158] (346×260 pixels), as well as camera poses and LiDAR data. Ground truth optical flow is computed as the motion field [179], given the camera velocity and the depth of the scene (from the LiDAR). We select several excerpts from the *outdoor_day1* sequence with a forward motion. This motion is reasonably well approximated by collapse-enabled warps such as (3.7). In total, we evaluate on 3.2 million events spanning 10 s.

The *DSEC* dataset [55] is a more recent driving dataset with a higher resolution event camera (Prophesee Gen3, 640×480 pixels). Ground truth optical flow is also computed as the motion field using the scene depth from a LiDAR [56]. We evaluate on the *zurich_city_11* sequence, using in total 380 million events spanning 40 s.

The *ECD* dataset [105] is the de-facto standard to assess event camera ego-motion [47,177,135,63,129,103,175]. Each sequence provides events, frames, a calibration file, and IMU data (at 1kHz) from a DAVIS240C camera [13] (240×180 pixels), as well as ground truth camera poses from a motion capture system (at 200Hz). For rotational motion estimation (3 DOFs), we use the natural-looking *boxes_rotation* and *dynamic_rotation* sequences. We evaluate on 43 million events (10 s) of the box sequence, and on 15 million events (11 s) of the dynamic sequence.

The driving datasets (MVSEC, DSEC) and the selected sequences in the ECD dataset have different type of motions: forward (which enables event collapse) vs. rotational (which does not suffer from event collapse). Each sequence serves a different test purpose, as discussed in the next sections.

Metrics. The metrics used to assess optical flow accuracy (MVSEC and DSEC datasets) are the Average Endpoint Error (AEE) and the percentage of pixels with AEE greater than N pixels (denoted by "NPE", for $N = \{3, 10, 20\}$). Both are measured over pixels with valid ground-truth values. We also use the FWL metric [155] to assess event alignment by means of the IWE sharpness (the FWL is the IWE variance relative to that of the identity warp).

Following previous works [45,113,63], rotational motion accuracy is assessed as the RMS error of angular velocity estimation. Angular velocity ω is assumed constant over a window of events, estimated and compared with the ground truth at the midpoint of the window. Additionally, we use the FWL metric to gauge event alignment [155].

The event time windows are as follows: the events in the time spanned by $dt = 4$ frames in MVSEC (standard in [180,56,65]), 500k events for DSEC, and 30k events for ECD [63]. The regularizer weights for divergence (λ_{div}) and deformation (λ_{def})are as follows: $\lambda_{\text{div}} = 2$ and $\lambda_{\text{def}} = 5$ for MVSEC, $\lambda_{\text{div}} = 50$ and $\lambda_{\text{def}} = 100$ for DSEC, and $\lambda_{\text{div}} = 5$ and $\lambda_{\text{def}} = 10$ for ECD experiments. Also the weights for the rate of the change (λ_{rate}) are $\lambda_{\text{rate}} = 0.2$ for MVSEC, and $\lambda_{\text{rate}} = 1.0$ for DSEC.

3.6.2 Effect of the Regularizers on Collapse-enabled Warps

Tables 3.1 and 3.2 report the results on the MVSEC and DSEC benchmarks, respectively, using two different loss functions G: the IWE variance (3.4) and the squared magnitude of the IWE gradient, abbreviated "Gradient Magnitude" [45]. For MVSEC, we report the accuracy within the time interval of $dt = 4$ grayscale frame (at ≈ 45Hz). The optimization algorithm is the Tree-structured Parzen Estimator (TPE) sampler [12] for both experiments, with number of sampling points equal to 300 (1 DOF) and 600 (4 DOFs). The tables quantitatively capture the collapse phenomenon suffered by the original CMax framework [46] and the whitening technique [113]. Their high FWL values indicate that contrast is maximized, however, the AEE and NPE values are exceedingly high (e.g., $>$ 80 pixels, 20PE $>$ 80%), indicating that the estimated flow is unrealistic.

Table 3.1: Results on MVSEC dataset[179]. The proposed regularizers are in bold. "RCAD" denotes the rate of change of area deformation. The best values per column per group are in bold, and second best are underlined. An asterisk in FWL indicates event collapse occurred.

		Variance					Gradient Magnitude				
		AEE ↓	3PE ↓	10PE ↓	20PE ↓	FWL ↑	AEE ↓	3PE ↓	10PE ↓	20PE ↓	FWL ↑
	Ground truth flow	–	–	–	–	1.05	–	–	–	–	1.05
	Identity warp	4.85	60.59	10.38	0.31	1.00	4.85	60.59	10.38	0.31	1.00
1 DOF	No regularizer	89.34	97.30	95.42	92.39	*1.90	85.77	93.96	86.24	83.45	*1.87
1 DOF	Whitening[113]	89.58	97.18	96.77	93.76	*1.90	81.10	90.86	89.04	86.20	*1.85
1 DOF	**Divergence**[145]	4.00	46.02	2.77	**0.05**	1.12	2.87	32.68	2.52	0.03	1.17
1 DOF	**Deformation**[145]	4.47	52.60	5.16	0.13	1.08	3.97	48.79	3.21	0.07	1.09
1 DOF	**Div. + Def.**[145]	3.30	**33.09**	**2.61**	0.48	**1.20**	**2.85**	**32.34**	**2.44**	**0.03**	**1.17**
1 DOF	**RCAD**[148]	**3.17**	36.65	4.01	0.10	1.16	3.02	34.47	3.40	0.07	1.17
4 DOF	No regularizer	90.22	90.22	96.94	93.86	*2.05	91.26	99.49	95.06	91.46	*2.01
4 DOF	Whitening[113]	90.82	99.11	98.04	95.04	*2.04	88.38	98.87	92.41	88.66	*2.00
4 DOF	**Divergence**[145]	7.25	81.75	18.53	0.69	1.09	5.37	66.18	10.81	**0.28**	1.14
4 DOF	**Deformation**[145]	8.13	87.46	18.53	1.09	1.03	5.25	64.79	13.18	0.37	1.15
4 DOF	**Div. + Def.**[145]	5.14	65.61	10.75	0.38	**1.16**	5.41	66.01	13.19	0.54	1.14
4 DOF	**RCAD**[148]	**4.36**	**58.63**	**6.56**	**0.16**	1.15	**4.30**	**54.27**	**5.61**	0.31	**1.17**

Table 3.2: Results on DSEC dataset[55]. Same notation as Tab. 3.1.

		Variance					Gradient Magnitude				
		AEE ↓	3PE ↓	10PE ↓	20PE ↓	FWL ↑	AEE ↓	3PE ↓	10PE ↓	20PE ↓	FWL ↑
	Ground truth flow	–	–	–	–	1.09	–	–	–	–	1.09
	Identity warp	5.84	60.45	16.65	3.40	1.00	5.84	60.45	16.65	3.40	1.00
1 DOF	No regularizer	156.13	99.88	99.33	98.18	*2.58	156.08	99.93	99.40	98.11	*2.58
1 DOF	Whitening[113]	156.18	99.95	99.51	98.26	*2.58	156.82	99.88	99.38	98.33	*2.58
1 DOF	**Divergence**[145]	12.49	69.86	20.78	6.66	1.43	5.47	63.48	14.66	**1.35**	1.34
1 DOF	**Deformation**[145]	9.01	68.96	18.86	4.77	**1.40**	5.79	64.02	16.11	2.75	**1.36**
1 DOF	**Div. + Def.**[145]	6.06	68.48	17.08	**2.27**	1.36	5.53	64.09	15.06	1.37	1.35
1 DOF	**RCAD**[148]	**5.81**	**57.19**	**14.73**	3.05	1.34	**5.31**	**54.85**	**14.17**	3.10	1.20
4 DOF	No regularizer	157.54	99.97	99.64	98.67	*2.64	157.34	99.94	99.53	98.44	*2.62
4 DOF	Whitening[113]	157.73	99.97	99.66	98.71	*2.60	156.12	99.91	99.26	97.93	*2.61
4 DOF	**Divergence**[145]	14.35	90.84	41.62	10.82	1.35	10.43	91.38	41.63	9.43	1.21
4 DOF	**Deformation**[145]	15.12	94.96	62.59	22.62	1.25	10.01	90.15	39.45	8.67	1.25
4 DOF	**Div. + Def.**[145]	**10.06**	**90.65**	**40.61**	**8.58**	1.26	10.39	91.02	41.81	9.40	1.23
4 DOF	**RCAD**[148]	11.51	91.50	42.29	11.05	**1.30**	**9.55**	**88.94**	**35.96**	**7.74**	**1.31**

By contrast, our regularizers ("Divergence", "Deformation", and "RCAD", which denotes Rate of change of area deformation) work well to mitigate the collapse, as observed in smaller AEE and *N*PE values. Compared with the values of no regularizer or whitening[113], our regularizers achieve more than 90% improvement for AEE on average. The AEE values

are high for optical flow standards ($4-8$ pix in MVSEC vs. $0.5-1$ pixel[180], or $10-20$ pix in DSEC vs. $2-5$ pix[56]), however, this is due to the fact that the warps used have very few DOFs (≤ 4) compared to the considerably higher DOFs ($2N_p$) of optical flow estimation algorithms. The same reason explains the high 3PE values (standard in[58]): using an end-point error threshold of 3 pix to consider that the flow is correctly estimated does not convey the intended goal of inlier/outlier classification for the low-DOF warps used. This is the reason why Tabs. 3.1 and 3.2 also report 10PE, 20PE metrics, and the values for the identity warp (zero flow). As expected, for the range of AEE values in the tables, the 10PE and 20PE figures demonstrate the large difference between methods suffering from collapse (20PE>80%) and those that do not (20PE<1.1% for MVSEC and <22.6% for DSEC).

The FWL values of our regularizers are moderately high (≥ 1), indicating that event alignment is better than that of the identity warp. However, since the FWL depends on the number of events[155], it is not easy to establish a global threshold to classify each method as suffering from collapse or not. The AEE, 10PE and 20PE are better for such a classification.

The collapse results are more visible in Fig. 3.9, where we used Divergence (3.9) and Deformation (3.5). Without a regularizer the events collapse in the MVSEC an DSEC sequences. Our regularizers successfully mitigate the event collapse, having a remarkable impact on the estimated motion.

The qualitative results for the RCAD (3.20) are shown in Fig. 3.10. Notice that the area deformation map (3.4) shows collapse *only* at pixels with warped events, while the RCAD regularizer provides dense maps (even in pixels with no events, corresponding to homogeneous brightness regions) because it is purely geometric, based on the motion parameters. All of the three proposed regularizers successfully mitigate event collapse, however, the computational complexity is different, which we investigate in Sec. 3.6.4.

3.6.3 Effect of the Regularizers on Well-posed Warps

Table 3.3 shows the results on the ECD dataset for a well-posed warp (3-DOF rotational motion, in the benchmark). We use the variance loss and the Adam optimizer[83] with 100 iterations. All values in the table (RMS error and FWL, with and without regularization) are very similar, indicating that: (*i*) our regularizers do not affect the motion estimation algorithm, (*ii*) results without regularization are good due to the well-posed warp. This is qualitatively shown in the bottom part of Fig. 3.9. The fluctuations of the divergence and deformation values away from those of the identity warp (0 and 1, respectively) are at least one order of magnitude smaller than the collapse-enabled warps (e.g., 0.2 vs. 2).

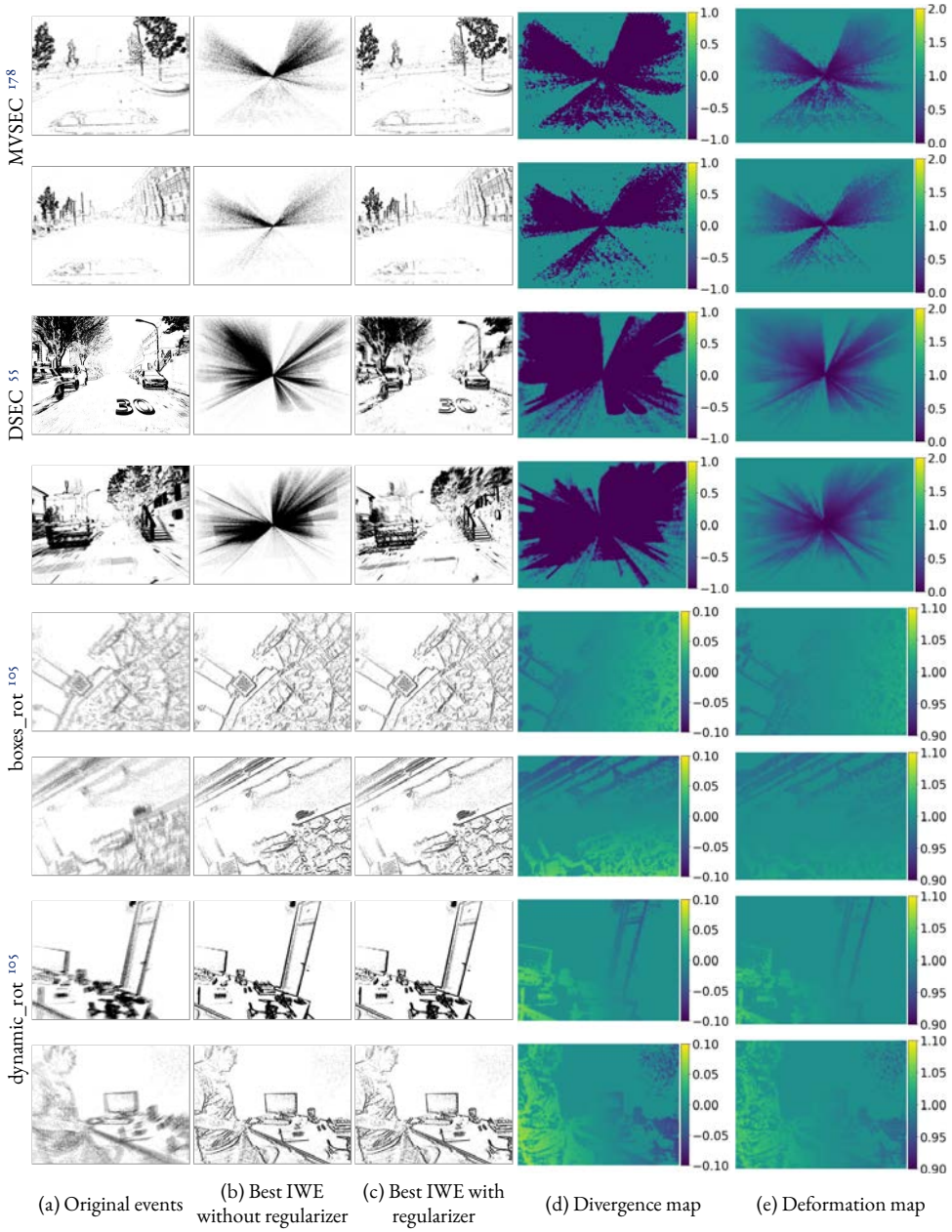


Figure 3.9: *Proposed regularizers and collapse analysis*. The scene motion is approximated by 1-DOF warp (zoom in/out) for MVSEC [178] and DSEC [55] sequences, and 3-DOF warp (rotation) for boxes and dynamic ECD sequences [105]. (a) Original events. (b) Best warp without regularization. Event collapse happens for 1-DOF warp. (c) Best warp with regularization. (d) Divergence map ((3.11) is zero-based). (e) Deformation map ((3.16), centered at 1). Our regularizers successfully penalize event collapse and do not damage non-collapsing scenarios.

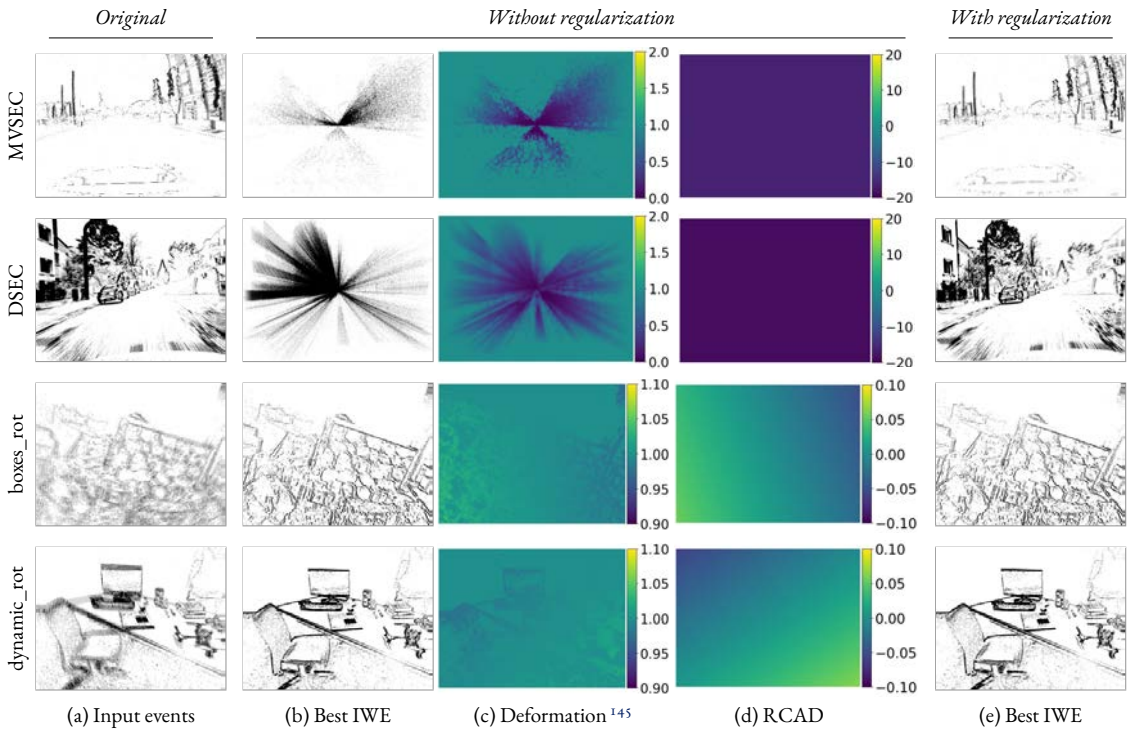


Figure 3.10: *Qualitative comparison between Deformation and Rate of change of area deformation ("RCAD").* (a) Original events. (b)-(d) Results without regularization: 1-DOF motion results (MVSEC [178] and DSEC [55]) are trapped in global optima of event collapse, as shown in the IWEs (b). The regularizers in such collapse cases (c)-(d) are very large compared with the well-posed warp cases (boxes_rot and dynamic_rot rows). (e) Results with the proposed regularizer: it mitigates collapse for MVSEC and DSEC scenes while it does not harm the ECD scenes. Best viewed in the electronic version.

Table 3.3: Results on ECD dataset [105].

	boxes_rot		dynamic_rot	
	RMS ↓	FWL ↑	RMS ↓	FWL ↑
Ground truth pose	–	1.559	–	1.414
No regularizer	8.858	1.562	4.823	1.420
Deformation [145]	8.664	1.561	4.822	1.420
Div. + Def. [145]	6.885	1.562	4.822	1.420
RCAD [148]	6.877	1.562	4.822	1.420

3.6.4 Runtime Comparison

Table 3.4 reports the runtime comparison of the methods, notably with respect to the original CMax ("No regularizer"). We use Python (3.9.12) on a CPU (Mac M1 2020, 8 Cores), and

Table 3.4: Comparison of runtime in milliseconds, averaged over 400 trials. MVSEC: 30k events. DSEC: 500k events.

	MVSEC		DSEC	
	Var.	Grad.	Var.	Grad.
No regularizer	**7.3**	**7.9**	**111.3**	**112.9**
Whitening [113]	8.2	8.4	205.5	206.9
Deformation [145]	20.2	21.1	304.4	307.6
Div. + Def. [145]	32.4	31.6	505.0	506.1
RCAD [148]	7.4	8.0	111.4	113.5

average the runtime over 400 trials. The whitening technique [113] is slower than the original CMax ("No regularizer"). The runtime difference is due to an extra SVD step on the events, which is more noticeable (2× slower) in the DSEC dataset than in MVSEC because it uses more events. The "Deformation" regularizer in [145] is also two to three times slower than the original CMax. When both regularizers are combined ("Div. + Def."), the runtime becomes even larger. Finally, the RCAD approach has almost the same runtime as the original CMax, since its complexity is $O(1)$, thus it is two to four times faster than competing methods.

Figure 3.11 visualizes the accuracy and runtime of the methods (on DSEC data). Runtime is reported relative to the "No regularizer" case. It clearly shows that the rate-of-change regularizer is the only effective approach against event collapse that does not compromise the speed of the CMax framework.

3.6.5 Sensitivity Analysis

The landscapes of loss functions as well as sensitivity analysis of λ are shown in Fig. 3.12, for the MVSEC experiments. Without regularizer ($\lambda = 0$), all objective functions tested (variance, gradient magnitude, and average timestamp [180]) suffer from event collapse, which is the undesired global minimum of (3.6). Reaching the desired local optimum depends on the optimizing algorithm and its initialization (e.g., starting gradient descent close enough to the local optimum). Our regularizers (Divergence and Deformation) change the landscape: the previously undesired global minimum becomes local, and the desired minimum becomes the new global one as λ increases.

Specifically, the larger the weight λ, the smaller the effect of the undesired minimum (at $h_z = 1$). However, this is true only within some reasonable range: a too large λ discards the data-fidelity part G in (3.6), which is unwanted because it would remove the desired local optimum (near $h_z \approx 0$). Minimizing (3.6) with only the regularizer is not sensible.

Observe that for completeness, we include the average timestamp loss in the last column.

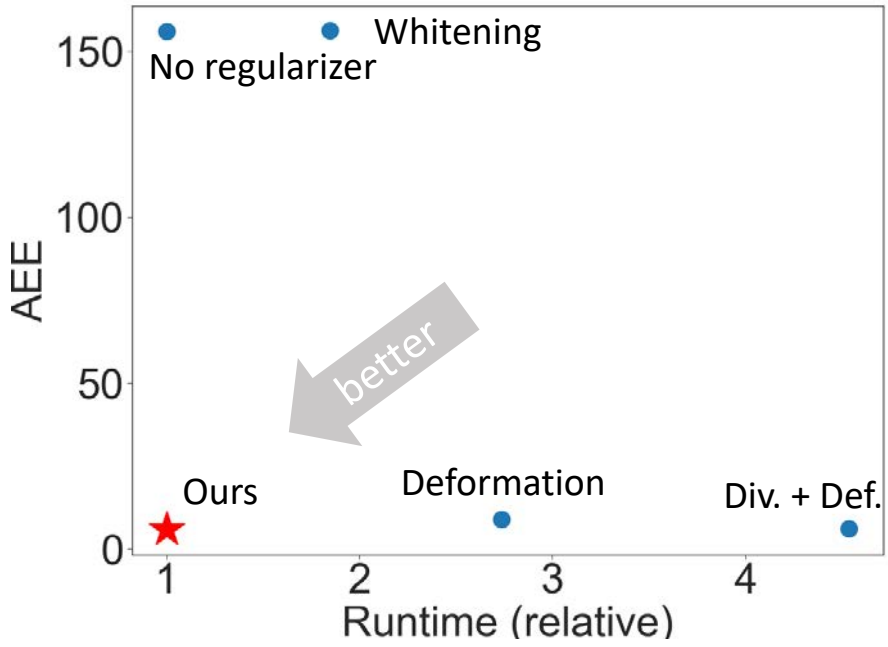


Figure 3.11: *Runtime comparison* for the DSEC experiment. Runtime is relative to that of the original CMax ("No regularizer"). The rate of change of area deformation (denoted as "Ours") regularizer has desirable properties: small AEE and runtime.

However, this loss also suffers from an undesired optimum in the expansion region ($h_z \approx -1$). Our regularizers could be modified to also remove this undesired optimum, but investigating this particular loss, which was proposed as an alternative to the original contrast loss, is outside the scope of this work.

3.6.6 Application to Motion Segmentation

While most of the results on standard datasets comprise stationary scenes, we have also provided results on a dynamic scene (from dataset[105]). Since the time spanned by each set of events processed is small, the scene motion is also small (even for complicated objects like the person in the bottom row of Fig. 3.9), hence often a single warp fits the scene reasonably well. In some scenarios, a single warp may not be enough to fit the event data because there are distinctive motions in the scene of equal importance. Our proposed regularizers can be extended to such more complex scene motions. To this end, we demonstrate it with an example in Fig. 3.13.

Specifically, we use the MVSEC dataset, in a clip where the scene consists of two motions: the ego-motion (forward motion of the recording vehicle) and the motion of a car driving in

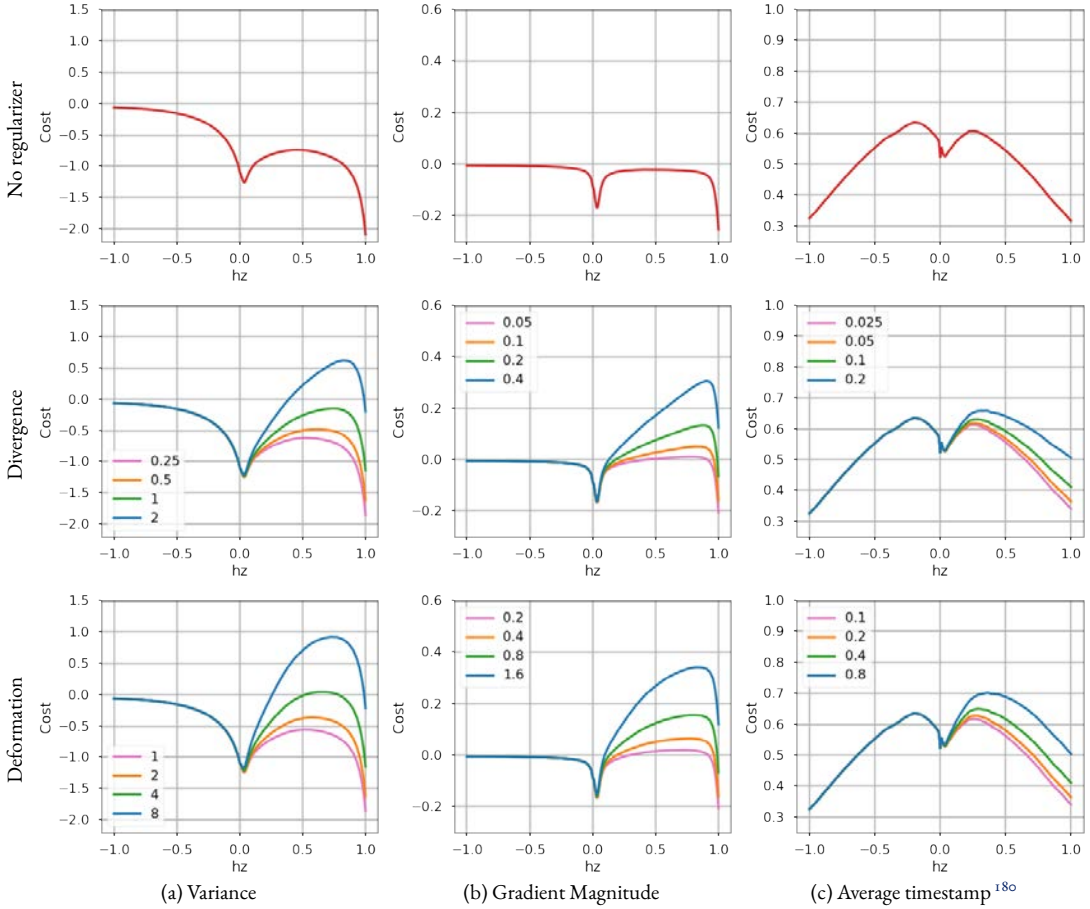
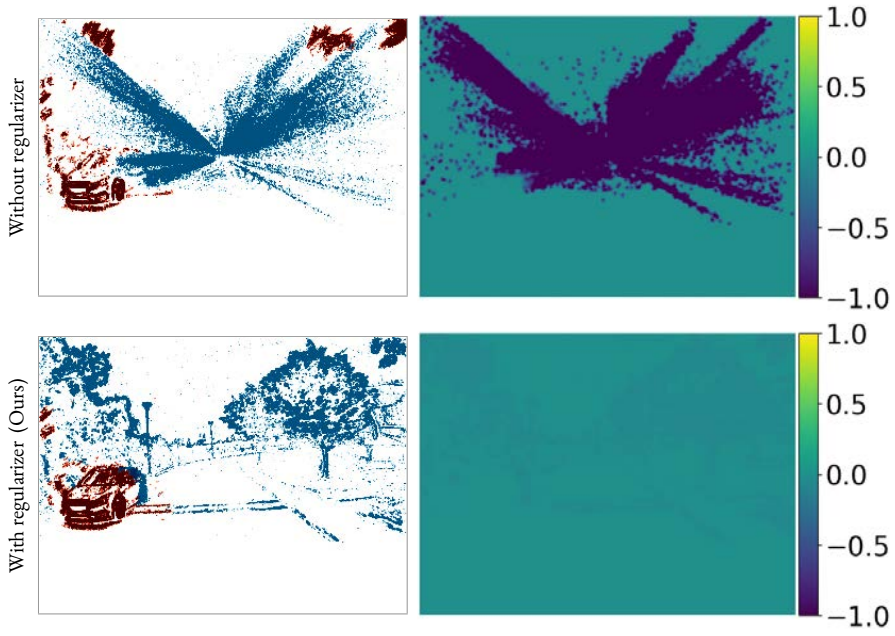


Figure 3.12: *Cost function landscapes* over the warp parameter h_z for: (a) Image variance [46], (b) Gradient Magnitude [45], and (c) Mean Square of Average Timestamp [180]. Data from MVSEC [178] with dominant forward motion. The legend weights denote λ in (3.6).

the opposite direction in a nearby lane (an independently moving object – IMO). We model the scene using the combination of two warps. Intuitively, the 1-DOF warp (3.7) describes the ego-motion, while the feature flow (2 DOFs) describes the IMO. Then, we apply the contrast maximization approach (augmented with our regularizing terms) and the expectation-maximization scheme in [152] to segment the scene, to determine which events belong to each motion. The results in Fig. 3.13 clearly show the effectiveness of our regularizer, even for such a commonplace and complex scene. Without regularizers, (*i*) event collapse appears in the ego-motion cluster of events and (*ii*) a considerable portion of the events that correspond to ego-motion are assigned to the second cluster (2-DOF warp), thus causing a segmentation



(a) IWE with segmentation (b) Divergence map

Figure 3.13: *Application to Motion Segmentation.* (a) Output IWE, whose colors (red and blue) represent different clusters of events (segmented according to motion). (b) Divergence map. The range of divergence values is larger in the presence of event collapse than in its absence. Our regularizer (divergence in this example) mitigates the event collapse for this complex motion, even with an independently moving object (IMO) in the scene.

failure. Our regularization approach mitigates event collapse (bottom row of Fig. 3.13) and provides the correct segmentation: the 1-DOF warp fits the ego-motion and the feature flow (2-DOF warp) fits the IMO.

3.6.7 Application: Time-to-contact

The parametrization of collapse-enabled warps has useful implications toward future application on intelligent vehicles, such as advanced driver-assistance system (ADAS). Let us introduce another interpretation of the parameter h_z. For a freely moving camera with linear and angular velocities $\mathbf{V}$ and ω, respectively, the apparent velocity $\mathbf{v}(\mathbf{x})$ on the image plane of a 3D point $\mathbf{X} = (x, y, Z(\mathbf{x}))^\top$ (at depth $Z(\mathbf{x})$ with respect to the camera) can be computed using the 2×6 feature sensitivity matrix[27]:

$$\mathbf{v}(\mathbf{x}) = \begin{pmatrix} \frac{-1}{Z(\mathbf{x})} & 0 & \frac{x}{Z(\mathbf{x})} & xy & -(1+x^2) & y \\ 0 & \frac{-1}{Z(\mathbf{x})} & \frac{y}{Z(\mathbf{x})} & 1+y^2 & -xy & x \end{pmatrix} \begin{pmatrix} \mathbf{V} \\ \omega \end{pmatrix}, \tag{3.29}$$

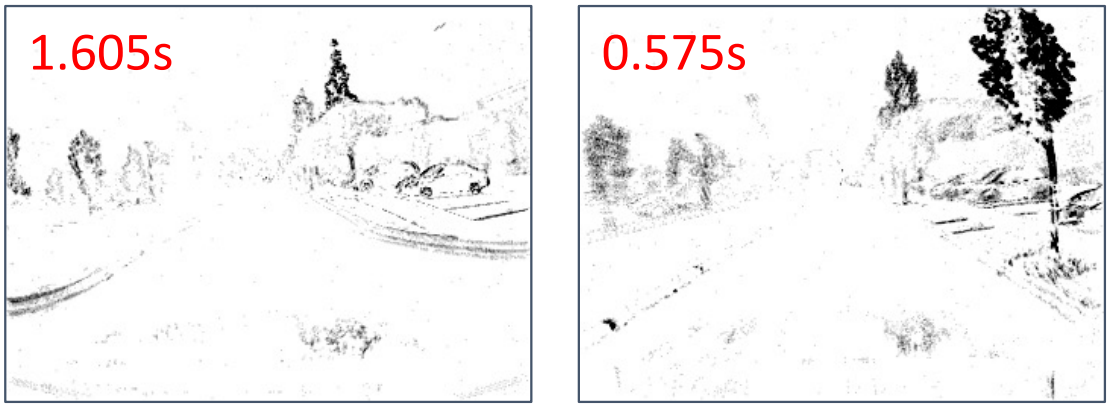


Figure 3.14: *Application of estimating Time to Contact.* The parametrization with h_z in the 1-DOF warp can be used to approximate the TTC for the dominant depth of the scene represented by the events (e.g., the trees).

which can be used to warp events:

$$\mathbf{x}'_k = \mathbf{x}_k - \mathbf{v}(\mathbf{x})t_k. \tag{3.30}$$

Assuming a vehicle with body-frame velocity v_z, i.e., $\mathbf{V} \equiv (0, 0, v_z)^\top$, $\omega \equiv (0, 0, 0)^\top$, the motion field (3.29) becomes $\mathbf{v}(\mathbf{x}) = (v_z/Z(\mathbf{x}))\,\mathbf{x}$, and substituting in (3.30) gives $\mathbf{x}'_k = (1 - v_z/Z(\mathbf{x}))t_k$. Comparing this expression to (3.7), and assuming $Z(\mathbf{x})$ is spatially invariant, we identify

$$h_z = \frac{v_z}{Z}, \tag{3.31}$$

i.e., the parameter h_z is inverse of the time-to-contact or time-to-collision (TTC)[21].

Figure 3.14 shows two examples of TTC from the MVSEC dataset. It is remarkable that this 1-DOF warp model can be related to the popular concept in ADAS, and our regularizer plays an important role toward real-time computation of TTC given its runtime. Also note that (3.31) establishes a relation between TTC, vehicle speed and scene depth, and that the TTC can be used to estimate the scene depth given the vehicle speed, or vice versa, the vehicle speed given the scene depth. We hope this connection helps future implementation of event-camera application in collision avoidance systems.

3.7 Conclusion

We have analyzed the event collapse phenomenon of the CMax framework and proposed three collapse metrics: divergence, area-based deformation, and rate of change of the deformation. Our experimental results on publicly available datasets demonstrate that the pro-

posed regularizers metrics mitigate the phenomenon for collapse-enabled warps while they do not harm well-posed warps. To the best of our knowledge, our regularizers are the only effective solution compared to the unregularized CMax framework and whitening. The proposed regularized CMax achieves, on average, more than 90% improvement on optical flow endpoint error calculation (AEE) on collapse-enabled warps. Furthermore, the runtime comparison demonstrate that the rate of change of area deformation does not trade-off the runtime of the original CMax framework, resulting in 2 to 4 times faster than other methods.

This is the first work that focuses on the paramount phenomenon of event collapse. No prior work has analyzed this phenomenon in such detail or proposed new regularizers without additional data or reparameterizing the search space[180,113,121]. As we analyzed various warps from 1 DOF to 4 DOFs, we hope that the ideas presented here inspire further research to tackle more complex warp models. For more complex warps, like those used in dense optical flow estimation[180,65], the divergence, area-based deformation, or rate of change of area deformation could be approximated using finite difference formulas.

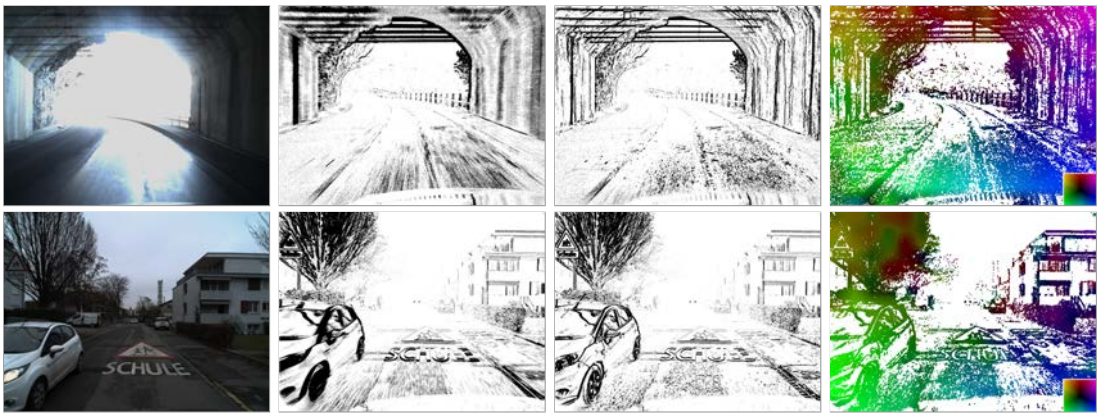
In this chapter, we focused on the low-DOF ego-motion estimation problems. We analyzed event collapse in the Contrast Maximization framework and proposed new regularizers that effectively and/or efficiently improve the landscape (well-posedness) of the problem, which has resulted in the extension of CMax to tackle more complex motions.

4
Optical Flow Estimation

4.1 Introduction

Event cameras are novel bio-inspired vision sensors that naturally respond to motion of edges in image space with high dynamic range (HDR) and minimal blur at high temporal resolution (on the order of µs)[125,10]. These advantages provide a rich signal for accurate motion estimation in difficult real-world scenarios for frame-based cameras. However such a signal is, by nature, asynchronous and sparse, which is not compatible with traditional computer vision algorithms. This poses the challenge of rethinking visual processing[43,86]: motion patterns (i.e., *optical flow*) are no longer obtained by analyzing the intensities of images captured at regular intervals, but by analyzing the stream of events (per-pixel brightness changes) produced by the event camera.

Multiple methods have been proposed for event-based optical flow estimation. They can be broadly categorized in two: (*i*) model-based methods, which investigate the principles and characteristics of event data that enable optical flow estimation, and (*ii*) learning-based methods, which exploit correlations in the data and/or apply the above-mentioned principles to compute optical flow. One of the challenges of event-based optical flow is the lack of ground truth flow in real-world datasets (at µs resolution and HDR)[43], which makes it difficult to evaluate and compare the methods properly, and to train supervised learning-based ones. Ground truth (GT) in de facto standard datasets[178,55] is given by the *motion field*[162] using additional depth sensors and camera information. However, such data is limited by



(a) Frame (reference) (b) Events (zero flow) (c) Warped events, IWE (d) Estimated flow

Figure 4.1: Two test sequences (interlaken_00_b, thun_01_a) from the DSEC dataset[55]. Our optical flow estimation method produces sharp images of warped events (IWE) despite the scene complexity, the large pixel displacement and the high dynamic range. The examples utilize 500k events on an event camera with 640×480 pixels.

the field-of-view (FOV) and resolution (spatial and temporal) of the depth sensor, which do not match those of event cameras. Hence, it is paramount to develop interpretable optical flow methods that exploit the characteristics of event data, and that do not need expensive-to-collect and error-prone ground truth.

Among prior work, Contrast Maximization (CM)[46,45] is a powerful framework that allows us to tackle multiple motion estimation problems (rotational motion[47,80,63], homographic motion[46,113,121], feature flow estimation[176,177,140,154], motion segmentation[99,152,174,117], and also reconstruction[46,128,173]). It maximizes an objective function (e.g., contrast) that measures the alignment of events caused by the same scene edge. The intuitive interpretation is to estimate the motion by recovering the sharp (motion-compensated) image of edge patterns that caused the events. Preliminary work on applying CM to estimate optical flow has reported a problem of overfitting to the data, producing undesired flows that warp events to few pixels or lines[180] (i.e., event collapse[145]). This issue has been tackled by changing the objective function, from contrast to the energy of an average timestamp image[180,118,65], but this loss is not straightforward to interpret and makes training difficult to converge[45].

Given the state-of-the-art performance of CM in low-DOF motion problems and its issues in more complex motions (dense flow), we think prior work may have rushed to use CM in unsupervised learning of complex motions. There is a gap in understanding how CM can be sensibly extended to estimate dense optical flow accurately. In this paper we fill this gap and learn a few "secrets" that are also applicable to overcome the issues of previous approaches.

We propose to extend CM for dense optical flow estimation via a tile-based approach covering the image plane. We present several distinctive contributions:

1. A *multi-reference* focus loss function to improve accuracy and discourage overfitting (Sec. 4.2.2).

2. A principled *time-aware flow* to better handle occlusions, formulating event-based optical flow as a transport problem via differential equations (Sec. 4.2.3).

3. A *multi-scale* approach on the raw events to improve convergence to the solution and avoid getting trapped in local optima (Sec. 4.2.4).

The results of our experimental evaluation are surprising: the above design choices are key to our simple, model-based tile-based method (Fig. 4.1) achieving the best accuracy among all state-of-the-art methods, including supervised-learning ones, on the de facto benchmark of MVSEC indoor sequences[179]. Since our method is interpretable and produces better event alignment than the ground truth flow, both qualitatively and quantitatively, the experiments also expose the limitations of the current "ground truth". Finally, experiments demonstrate that the above key choices are transferable to unsupervised learning methods, thus guiding future design and understanding of more proficient Artificial Neural Networks (ANNs) for event-based optical flow estimation.

Because of the above, we believe that the proposed design choices deserve to be called "secrets"[157]. To the best of our knowledge, they are novel in the context of event-based optical flow estimation, e.g., no prior work considers constant flow along its characteristic lines, designs the multi-reference focus loss to tackle overfitting, or has explicitly defined multi-scale (i.e., multi-resolution) contrast maximization on the raw events.

4.2 Method

4.2.1 Event Cameras and Contrast Maximization

Event cameras have independent pixels that operate continuously and generate "events" $e_k \doteq (\mathbf{x}_k, t_k, p_k)$ whenever the logarithmic brightness at the pixel increases or decreases by a predefined amount, called contrast sensitivity. Each event e_k contains the pixel-time coordinates $(\mathbf{x}_k, t_k)$ of the brightness change and its polarity $p_k = \{+1, -1\}$. Events occur asynchronously and sparsely on the pixel lattice, with a variable rate that depends on the scene dynamics.

The CM framework[46] assumes events $\mathcal{E} \doteq \{e_k\}_{k=1}^{N_e}$ are generated by moving edges, and transforms them geometrically according to a motion model $\mathbf{W}$, producing a set of warped events $\mathcal{E}'_{t_{\text{ref}}} \doteq \{e'_k\}_{k=1}^{N_e}$ at a reference time t_{ref}:

$$e_k \doteq (\mathbf{x}_k, t_k, p_k) \;\; \mapsto \;\; e'_k \doteq (\mathbf{x}'_k, t_{\text{ref}}, p_k). \tag{4.1}$$

The warp $\mathbf{x}'_k = \mathbf{W}(\mathbf{x}_k, t_k; \theta)$ transports each event from t_k to t_{ref} along the motion curve that passes through it. The vector θ parametrizes the motion curves. Transformed events are aggregated on an image of warped events (IWE):

$$I(\mathbf{x}; \mathcal{E}'_{t_{\text{ref}}}, \theta) \doteq \sum_{k=1}^{N_e} \delta(\mathbf{x} - \mathbf{x}'_k), \tag{4.2}$$

where each pixel $\mathbf{x}$ sums the number of warped events $\mathbf{x}'_k$ that fall within it. The Dirac delta δ is approximated by a Gaussian, $\delta(\mathbf{x}-\mu) \approx \mathcal{N}(\mathbf{x}; \mu, \varepsilon^2 \mathtt{Id})$ with $\varepsilon = 1$ pixel. Next, an objective function $f(\theta)$ is built from the transformed events, such as the contrast of the IWE (4.2), given by the variance

$$\text{Var}\big(I(\mathbf{x}; \theta)\big) \doteq \tfrac{1}{|\Omega|} \int_\Omega (I(\mathbf{x}; \theta) - \mu_I)^2 d\mathbf{x}, \tag{4.3}$$

with mean $\mu_I \doteq \frac{1}{|\Omega|} \int_\Omega I(\mathbf{x}; \theta) d\mathbf{x}$. The objective function measures the goodness of fit between the events and the candidate motion curves (warp). Finally, an optimization algorithm iterates the above steps until convergence. The goal is to find the motion parameters that maximize the alignment of events caused by the same scene edge. Event alignment is measured by the strength of the edges of the IWE, which is directly related to image contrast[61].

Dense optical flow. In the task of interest, the warp used is[180,118,65]:

$$\mathbf{x}'_k = \mathbf{x}_k + (t_k - t_{\text{ref}})\, \mathbf{v}(\mathbf{x}_k), \tag{4.4}$$

where $\theta = \{\mathbf{v}(\mathbf{x})\}_{\mathbf{x}\in\Omega}$ is a flow field on the image plane at a set time, e.g., t_{ref}.

4.2.2 Multi-reference Focus Objective Function

Zhu et al.[180] report that the contrast objective (variance) overfits to the events. This is in part because the warp (4.4) can describe very complex flow fields, which can push the events to accumulate in few pixels[145]. To mitigate overfitting, we reduce the complexity of the flow field by dividing the image plane into a tile of non-overlapping patches, defining a flow vector at the center of each patch and interpolating the flow on all other pixels (we show the tiles in Sec. 4.2.4).

However, this is not enough. Additionally, we discover that warps that produce sharp IWEs *at any* reference time t_{ref} have a regularizing effect on the flow field, discouraging overfitting. This is illustrated in Fig. 4.2. In practice we compute the *multi-reference* focus loss using 3 reference times: t_1 (min), $t_{\text{mid}} \doteq (t_1 + t_{N_e})/2$ (midpoint) and t_{N_e} (max). The flow field is defined only at one reference time.

Furthermore, we measure event alignment using the magnitude of the IWE gradient because: (*i*) it has top accuracy performance among the objectives in[45], (*ii*) it is sensitive to the arrangement (i.e., permutation) of the IWE pixel values, whereas the variance of the IWE

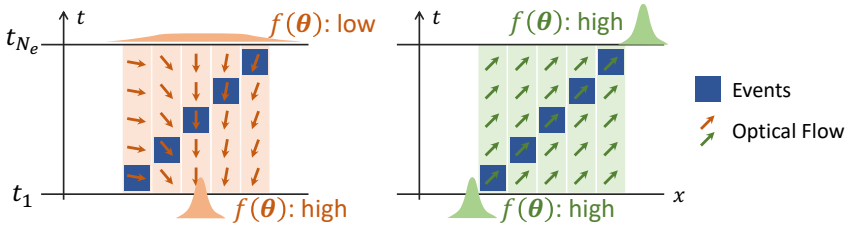


Figure 4.2: *Multi-reference focus loss.* Assume an edge moves from left to right. Flow estimation with single reference time (t_1) can overfit to the data, warping all events into a single pixel, which results in a maximum contrast (at t_1). However, the same flow would produce low contrast (i.e., a blurry image) if events were warped to time t_{N_e}. Instead, we favor flow fields that produce high contrast (i.e., sharp images) at any reference time (here, $t_{\text{ref}} = t_1$ and $t_{\text{ref}} = t_{N_e}$). See results in Fig. 4.7.

(4.3) is not, (*iii*) it converges more easily than other objectives we tested, (*iv*) it differs from the Flow Warp Loss (FWL)[155], which is defined using the variance (4.3) and will be used for evaluation.

Finally, letting the (squared) gradient magnitude of the IWE be

$$G(\theta; t_{\text{ref}}) \doteq \frac{1}{|\Omega|} \int_\Omega \|\nabla I(\mathbf{x}; t_{\text{ref}})\|^2 \, d\mathbf{x}, \tag{4.5}$$

the proposed multi-reference focus objective function becomes the average of the G functions of the IWEs at multiple reference times:

$$f(\theta) \doteq \left(G(\theta; t_1) + 2G(\theta; t_{\text{mid}}) + G(\theta; t_{N_e})\right) / \, 4G(0; -), \tag{4.6}$$

normalized by the value of the G function with zero flow (identity warp). The normalization in (4.6) provides the same interpretation as the FWL: $f < 1$ implies the flow is worse than the zero flow baseline, whereas $f > 1$ means that the flow produces sharper IWEs than the baseline.

Remark: Warping to two reference times (min and max) was proposed in[180], but with important differences: (*i*) it was done for the average timestamp loss, hence it did not consider the effect on contrast or focus functions[45], and (*ii*) it had a completely different motivation: to lessen a back-propagation scaling problem, so that the gradients of the loss would not favor events far from t_{ref}.

4.2.3 Time-aware Flow

State-of-the-art event-based optical flow approaches are based on frame-based ones, and so they use the warp (4.4), which defines the flow $\mathbf{v}(\mathbf{x})$ as a function of $\mathbf{x}$ (i.e., a pixel displacement between two given frames). However, this does not take into account the space-time

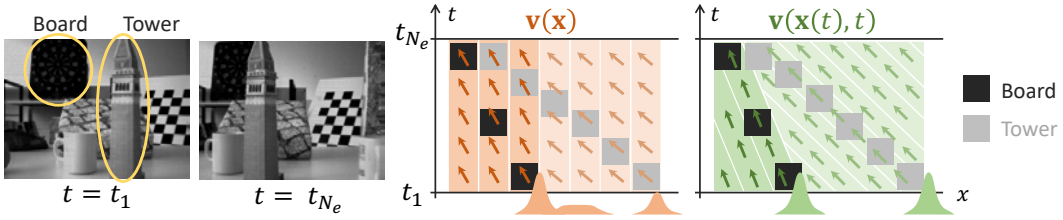


Figure 4.3: *Time-aware Flow.* Traditional flow (4.4), inherited from frame-based approaches, assumes per-pixel constant flow $\mathbf{v}(\mathbf{x}) = \text{const}$, which cannot handle occlusions properly. The proposed space-time flow assumes constancy along streamlines, $\mathbf{v}(\mathbf{x}(t), t) = \text{const}$, which allows us to handle occlusions more accurately. (See results in Fig. 4.8)

nature of events, which is the basis of the CM approach, because not all events at a pixel $\mathbf{x}_0$ are triggered at the same timestamp t_k. They do not need to be warped with the same velocity $\mathbf{v}(\mathbf{x}_0)$. Figure 4.3 illustrates this with an occlusion example taken from the slider_depth sequence [105]. Instead of $\mathbf{v}(\mathbf{x})$, the *event-based flow* should be a function of space-time, $\mathbf{v}(\mathbf{x}, t)$, i.e, *time-aware*, and each event e_k should be warped according to the flow defined at $(\mathbf{x}_k, t_k)$. Let us propose a more principled warp than (4.4).

To define a space-time flow $\mathbf{v}(\mathbf{x}, t)$ that is compatible with the propagation of events along motion curves, we are inspired by the method of characteristics [38]. Just like the brightness constancy assumption states that brightness is constant along the true motion curves in image space, we assume the flow is constant along its streamlines: $\mathbf{v}(\mathbf{x}(t), t) = \text{const}$ (Fig. 4.3). Differentiating in time and applying the chain rule gives a system of partial differential equations (PDEs):

$$\frac{\partial \mathbf{v}}{\partial \mathbf{x}} \frac{d\mathbf{x}}{dt} + \frac{\partial \mathbf{v}}{\partial t} = 0, \tag{4.7}$$

where, as usual, $\mathbf{v} = d\mathbf{x}/dt$ is the flow. The boundary condition is given by the flow at say $t = 0$: $\mathbf{v}(\mathbf{x}, 0) = \mathbf{v}^0(\mathbf{x})$. This system of PDEs essentially states how to propagate (i.e., *transport*) a given flow $\mathbf{v}^0(\mathbf{x})$, from the boundary $t = 0$ to the rest of space $\mathbf{x}$ and time t. The PDEs have advection terms and others that resemble those of the inviscid Burgers' equation [38] since the flow is transporting itself. We parametrize the flow at $t = t_{\text{mid}}$ (boundary condition), and then propagate it to the volume that encloses the current set of events $\mathcal{E}$. We develop two explicit methods to solve the PDEs, one with upwind differences and one with a conservative scheme adapted to Burgers' terms [141]. Each event e_k is then warped according to a flow $\hat{\mathbf{v}}$ given by the solution of the PDEs at $(\mathbf{x}_k, t_k)$:

$$\mathbf{x}'_k = \mathbf{x}_k + (t_k - t_{\text{ref}})\, \hat{\mathbf{v}}(\mathbf{x}_k, t_k). \tag{4.8}$$

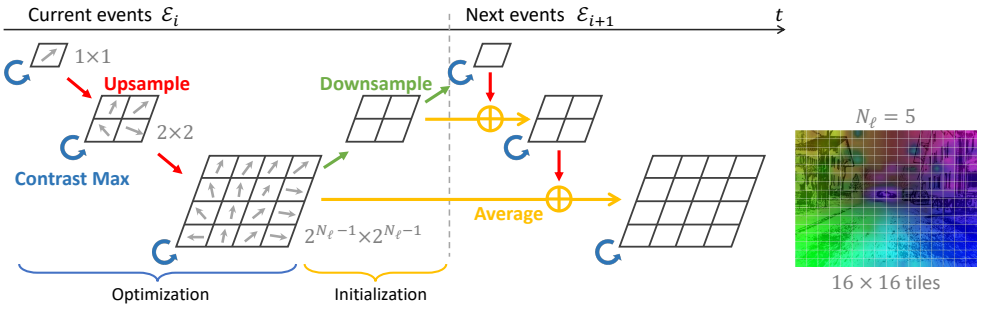


Figure 4.4: *Multi-scale Approach* using tiles (rectangles) and raw events.

4.2.4 Multi-scale Approach

Inspired by classical estimation methods, we combine our tile-based approach with a multi-scale strategy. The goal is to improve the convergence of the optimizer in terms of speed and robustness (i.e., avoiding local optima).

Some learning-based works [179,180,118] also have a multi-scale component, inherited from the use of a U-Net architecture. However, they work on discretized event representations (voxel grid, etc.) to be compatible with CNNs. In contrast, our tile-based approach works directly on raw events, without discarding or quantizing the temporal information in the event stream. While some prior work outside the context of optical flow has considered multi-resolution on raw events [90], there is no agreement on the best way to perform multi-resolution due to the sparse and asynchronous nature of events.

Our multi-scale CM approach is illustrated in Fig. 4.4. For an event set $\mathcal{E}_i$, we apply the tile-based CM in a coarse-to-fine manner (e.g., $N_\ell = 5$ scales). There are $2^{l-1} \times 2^{l-1}$ tiles at the l-th scale. We use bilinear interpolation to upscale between any two scales. If there is a subsequent set $\mathcal{E}_{i+1}$, the flow estimated from $\mathcal{E}_i$ is used to initialize the flow for $\mathcal{E}_{i+1}$. This is done by downsampling the finest flow to coarser scales. The coarsest scale initializes the flow for $\mathcal{E}_{i+1}$. For finer scales, initialization is computed as the average of the upsampled flow from the coarser scale of $\mathcal{E}_{i+1}$ and the same-scale flow from $\mathcal{E}_i$.

Composite Objective. To encourage additional smoothness of the flow, even in regions with few events, we include a flow regularizer $\mathcal{R}(\theta)$. The flow is obtained as the solution to the problem with the composite objective:

$$\theta^* = \arg\min_\theta \big(1/f(\theta) + \lambda \mathcal{R}(\theta)\big), \tag{4.9}$$

where, $\lambda > 0$ is the regularizer weight, and we use the total variation (TV) [136] as regularizer. We choose $1/f$ instead of simply $-f$ because it is convenient for ANN training, as we will apply in Sec 4.3.7.

4.3 Experiments

4.3.1 Datasets, Metrics and Hyper-parameters

We evaluate our method on sequences from the MVSEC dataset [178,179], which is the de facto dataset used by prior works to benchmark optical flow. It provides events, grayscale frames, IMU data, camera poses, and scene depth from a LiDAR [178]. The dataset was extended in [179] to provide ground truth optical flow, computed as the motion field [162] given the camera velocity and the depth of the scene. The event camera has 346×260 pixel resolution [158]. In total, we evaluate on 63.5 million events spanning 265 seconds.

We also evaluate on a recent dataset that provides ground truth flow: DSEC [56]. It consists of sequences recorded with Prophesee Gen3 event cameras, of higher resolution (640×480 pixels), mounted on a car. Optical flow is also computed as the motion field, with the scene depth from a LiDAR. In total, we evaluate on 3 billion events spanning the 208 seconds of the test sequences.

The metrics used to assess optical flow accuracy are the average endpoint error (AEE) and the percentage of pixels with AEE greater than 3 pixels (denoted by "% Out"), both are measured over pixels with valid ground-truth and at least one event in the evaluation intervals. We also use the FWL metric (the IWE variance relative to that of the identity warp) to assess event alignment [155].

In all experiments our method uses $N_\ell = 5$ resolution scales, $\lambda = 0.0025$ in (4.9), and the Newton-CG optimization algorithm with a maximum of 20 iterations/scale. The flow at t_{mid} is transported to each side via the upwind or Burgers' PDE solver (using 5 bins for MVSEC, 40 for DSEC), and used for event warping (4.8) (see Suppl. Mat.). In the optimization, we use 30k events for MVSEC indoor sequences, 40k events for outdoors, and 1.5M events for DSEC.

4.3.2 Results on MVSEC

Table 4.1 reports the results on the MVSEC benchmark. The different methods (rows) are compared on three indoor sequences and one outdoor sequence (columns). This is because many learning-based methods train on the other outdoor sequence, which is therefore not used for testing. Following Zhu et al., outdoor_day1 is tested only on specified 800 frames [179]. The top part of Tab. 4.1 reports the flow corresponding to a time interval of $dt = 1$ grayscale frame (at ≈ 45Hz, i.e., 22.2ms), and the bottom part corresponds to $dt = 4$ frames (89ms).

Our methods provide the best results among all methods in all indoor sequences and are the best among the unsupervised and model-based methods in the outdoor sequence. The errors for $dt = 4$ are about four times larger than those for $dt = 1$, which is sensible given the ratio of time interval sizes. We observe no significant differences between the three versions

Table 4.1: Results on MVSEC dataset[179]. Methods are sorted according to how much data they need: supervised learning (SL) requires ground truth flow; semi-supervised learning (SSL) uses grayscale images for supervision; unsupervised learning (USL) uses only events; and model-based (MB) needs no training data. Bold is the best among all methods; underlined is second best. Nagata et al.[109] evaluate on shorter time intervals; for comparison, we scale the errors to $dt = 1$.

		indoor_flying1		indoor_flying2		indoor_flying3		outdoor_day1	
	$dt = 1$	AEE ↓	%Out ↓	AEE ↓	%Out ↓	AEE ↓	%Out ↓	AEE ↓	%Out ↓
SL	EV-FlowNet-EST[52]	0.97	0.91	1.38	8.20	1.43	6.47	–	–
SL	EV-FlowNet+[155]	0.56	1.00	0.66	1.00	0.59	1.00	0.68	0.99
SL	E-RAFT[56]	–	–	–	–	–	–	**0.24**	1.70
SSL	EV-FlowNet (original)[179]	1.03	2.20	1.72	15.10	1.53	11.90	0.49	0.20
SSL	Spike-FlowNet[87]	0.84	–	1.28	–	1.11	–	0.49	–
SSL	Ziluo et al.[34]	0.57	0.10	0.79	1.60	0.72	1.30	0.42	**0.00**
USL	EV-FlowNet[180]	0.58	**0.00**	1.02	4.00	0.87	3.00	0.32	**0.00**
USL	EV-FlowNet (retrained)[118]	0.79	1.20	1.40	10.90	1.18	7.40	0.92	5.40
USL	FireFlowNet[118]	0.97	2.60	1.67	15.30	1.43	11	1.06	6.60
USL	ConvGRU-EV-FlowNet[65]	0.60	0.51	1.17	8.06	0.93	5.64	0.47	0.25
MB	Nagata et al.[109]	0.62	–	0.93	–	0.84	–	0.77	–
MB	Akolkar et al.[1]	1.52	–	1.59	–	1.89	–	2.75	–
MB	Brebion et al.[15]	0.52	0.10	0.98	5.50	0.71	2.10	0.53	0.20
MB	Ours (w/o time aware)	**0.42**	0.09	**0.60**	**0.59**	**0.50**	0.29	0.30	0.11
MB	Ours (Upwind)	**0.42**	0.10	**0.60**	**0.59**	**0.50**	**0.28**	0.30	0.10
MB	Ours (Burgers')	**0.42**	0.10	**0.60**	**0.59**	**0.50**	**0.28**	0.30	0.10
	$dt = 4$								
SSL	EV-FlowNet (original)[179]	2.25	24.70	4.05	45.30	3.45	39.70	1.23	7.30
SSL	Spike-FlowNet[87]	2.24	–	3.83	–	3.18	–	1.09	–
SSL	Ziluo et al.[34]	1.77	14.70	2.52	**26.10**	2.23	22.10	**0.99**	**3.90**
USL	EV-FlowNet[180]	2.18	24.20	3.85	46.80	3.18	47.80	1.30	9.70
USL	ConvGRU-EV-FlowNet[65]	2.16	21.51	3.90	40.72	3.00	29.60	1.69	12.50
MB	Ours (w/o time aware)	**1.68**	**12.79**	**2.49**	26.31	**2.06**	**18.93**	1.25	9.19
MB	Ours (Upwind)	1.69	12.83	**2.49**	26.37	**2.06**	19.02	1.25	9.23
MB	Ours (Burgers')	1.69	12.95	**2.49**	26.35	**2.06**	19.03	1.25	9.21

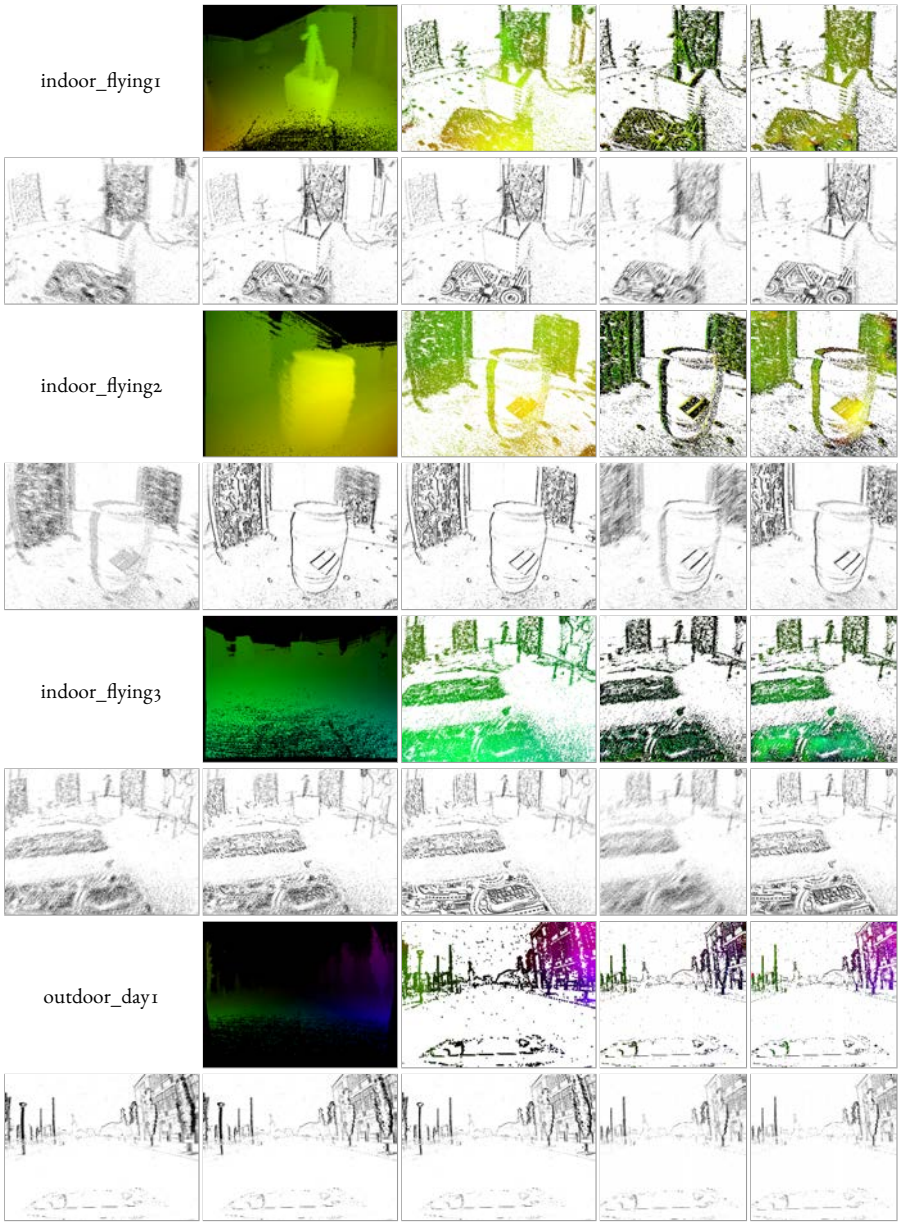


Figure 4.5: *MVSEC comparison* ($dt = 4$) of our method and two state-of-the-art baselines: ConvGRU-EV-FlowNet (USL)[65] and EV-FlowNet (SSL)[179]. For each sequence, the upper row shows the flow masked by the input events, and the lower row shows the IWE using the flow. Our method produces the sharpest motion-compensated IWEs. Note that learning-based methods crop input events to center 256×256 pixels, whereas our method does not. Black points in ground truth (GT) flow maps indicate the absence of LiDAR measurements. The optical flow color wheel is in Fig. 4.1.

of the method tested (warp models, see also Sec. 4.3.5), which can be attributed to the fact that the MVSEC dataset does not comprise large pixel displacements or occlusions.

Qualitative results are shown in Fig. 4.5, where we compare our method against the state of the art. Our method provides sharper IWEs than the baselines, without overfitting, and the estimated flow resembles the ground truth one.

Ground truth (GT) is not available on the entire image plane (see Fig. 4.5), such as in pixels not covered by the LiDAR's range, FOV, or spatial sampling. Additionally, there may be interpolation issues in the GT, since the LiDAR works at 20 Hz and the GT flow is given at frame rate (45 Hz). In the outdoor sequences, the GT from the LiDAR and the camera motion cannot provide correct flow for independently moving objects (IMOs). These issues of the GT are noticeable in the IWEs: they are not as sharp as expected. In contrast, the IWEs produced by our method are sharp. Taking now into account the GT quality on the comparison Table 4.1, it is remarkable that our method outperforms the state-of-the-art baselines on the indoor sequences, where GT has the best quality (with more points in the valid LiDAR range and no IMOs).

4.3.3 Results on DSEC

Table 4.2 gives quantitative results on all the DSEC Optical Flow benchmark. No GT flow is available for these sequences. Currently only the method that proposed the benchmark reports values[56]. As expected, this supervised learning method is better than ours in terms of flow accuracy because (*i*) it has additional training information (GT labels), and (*ii*) it is trained using the same type of GT signal used in the evaluation. Nevertheless, our method provides competitive results and is better in terms of FWL, which exposes similar GT quality issues as those of MVSEC: pixels without GT (LiDAR's FOV and IMOs). Qualitative results are shown in Fig. 4.6. Our method provides sharp IWEs, even for IMOs (car) and the road close to the camera. The FWL is computed using the same 100ms intervals used for the accuracy benchmark calculation. Since the FWL is sensitive to the number of events, the previous convention is consistent with the benchmark.

We observe that the evaluation intervals (100ms) are large for optical flow standards. In the benchmark, 80% of the GT flow has up to 22px displacement, which means that 20% of the GT flow is larger than 22px (on VGA resolution). The apparent motion during such intervals is sufficiently large that it breaks the classical assumption of scene points flowing in linear trajectories.

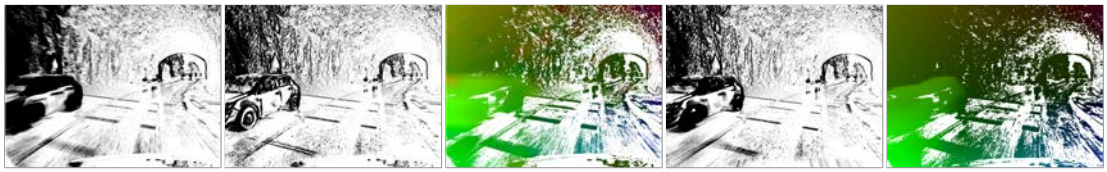
4.3.4 Effect of the Multi-reference Focus Loss

The effect of the proposed multi-reference focus loss is shown in Fig. 4.7. The single-reference focus loss function can easily overfit to the only reference time, pushing all events into a small

Table 4.2: Results on the DSEC optical flow benchmark[56].

	All			interlaken_00_b			interlaken_01_a			thun_01_a		
	AEE ↓	%Out ↓	FWL ↑	AEE ↓	%Out ↓	FWL ↑	AEE ↓	%Out ↓	FWL ↑	AEE ↓	%Out ↓	FWL ↑
E-RAFT[56]	0.79	2.68	1.29	1.39	6.19	1.32	0.90	3.91	1.42	0.65	1.87	1.20
Ours	3.47	30.86	**1.37**	5.74	38.93	**1.50**	3.74	31.37	**1.51**	2.12	17.68	**1.24**

	thun_01_b			zurich_city_12_a			zurich_city_14_c			zurich_city_15_a		
	AEE ↓	%Out ↓	FWL ↑	AEE ↓	%Out ↓	FWL ↑	AEE ↓	%Out ↓	FWL ↑	AEE ↓	%Out ↓	FWL ↑
E-RAFT[56]	0.58	1.52	1.18	0.61	1.06	1.12	0.71	1.91	1.47	0.59	1.30	1.34
Ours	2.48	23.56	**1.24**	3.86	43.96	**1.14**	2.72	30.53	**1.50**	2.35	20.99	**1.41**



(a) Events (b) IWE (Ours) (c) Flow (Ours) (d) IWE (SL)[56] (e) Flow (SL)[56]

Figure 4.6: *DSEC results* on the interlaken_00_b test sequence (no GT available). Since GT is missing at IMOs and points outside the LiDAR's FOV, the supervised method[56] may provide inaccurate predictions around IMOs and road points close to the camera, whereas our method produces sharp edges. For visualization, we use 1M events.

region of the image at t_1 while producing blurry IWEs at other times (t_{mid} and t_{N_e}). Instead, our proposed multi-reference focus loss discourages such overfitting, as the loss favors flow fields which produce sharp IWEs at *any* reference time. The difference is also noticeable in the flow: the flow from the single-reference loss is irregular, with a lot of spatial variability in terms of directions (many colors, often in opposite directions of the color wheel). In contrast, the flow from the multi-reference loss is considerably more regular.

4.3.5 Effect of the Time-Aware Flow

To assess the effect of the proposed time-aware warp (4.8), we conducted experiments on MVSEC, DSEC and ECD[105] datasets. Accuracy results are already reported in Tabs. 4.1 and 4.2. We now report values of the FWL metric in Tab. 4.3. For MVSEC, $dt = 1$ is a very short time interval, with small motion and therefore few events, hence the sharpness of the IWE with or without motion compensation are about the same (FWL $\approx$ 1). Instead, $dt = 4$ provides more events, and larger FWL values (1.1–1.3), which means that the contrast of the motion-compensated IWE is better than that of the zero flow baseline. All three methods provide sharper IWEs than ground truth. The advantages of the time-aware warp

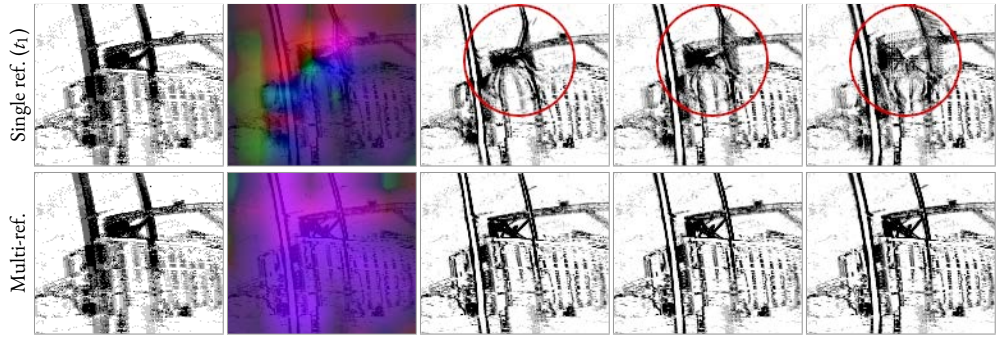


Figure 4.7: *Effect of the multi-reference focus loss.*

Table 4.3: FWL (IWE sharpness) results on MVSEC, DSEC, and ECD. Higher is better.

	MVSEC ($dt = 4$)				ECD	DSEC	
	indoor1	indoor2	indoor3	outdoor1	slider_depth	thun_00_a	zurich_city_07_a
Ground truth	1.09	1.20	1.12	1.07	–	1.01	1.04
Ours: w/o time aware	**1.17**	**1.30**	**1.23**	**1.11**	1.88	1.39	1.57
Ours: Upwind	**1.17**	**1.30**	**1.23**	**1.11**	1.92	1.40	1.60
Ours: Burgers'	**1.17**	**1.30**	**1.23**	**1.11**	**1.93**	**1.42**	**1.63**

(4.8) over (4.4) to produce better IWEs (higher FWL) are most noticeable on sequences like slider_depth [105] and DSEC (see Fig. 4.8) because of the occlusions and larger motions. Notice that FWL differences below 0.1 are significant [155], demonstrating the efficacy of time-awareness.

4.3.6 Effect of the Multi-scale Approach

The effect of the proposed multi-scale approach (Fig. 4.4) is shown in Fig. 4.9. This experiment compares the results of using multi-scale approaches (in a coarse-to-fine fashion) versus using a single (finest) scale. With a single scale, the optimizer gets stuck in a local extremal, yielding an irregular flow field (see the optical flow rows), which may produce a blurry IWE (e.g., outdoor_day1 scene). With three scales (finest tile and two downsampled ones), the flow becomes less irregular than with one single scale, but there may be regions with few events where the flow is difficult to estimate. With five scales the flow becomes smoother, more coherent over the whole image domain, while still being able to produce sharp IWEs.

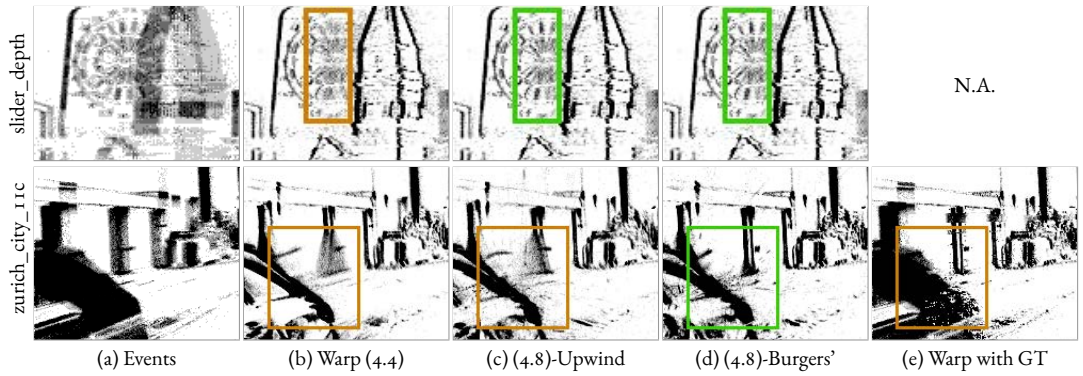


Figure 4.8: *Time-aware flow.* Comparison between 3 versions of our method: Burgers', upwind, and no time-aware (4.4). At occlusions (dartboard in slider_depth [105] and garage door in DSEC [55]), upwind and Burgers' produce sharper IWEs. Due to the smoothness of the flow conferred by the tile-based approach, some small regions are still blurry.

4.3.7 Application to Deep Neural Networks (DNN)

The proposed secrets are not only applicable to model-based methods, but also to unsupervised-learning methods. We train EV-FlowNet [179] in an unsupervised manner, using (4.9) as data-fidelity term and a Charbonnier loss [19] as the regularizer. Since the time-aware flow does not have a significant influence on the MVSEC benchmark (Tab. 4.1), we do not port it to the learning-based setting. We convert 40k events into the voxel-grid representation [180] with 5 time bins. The network is trained for 50 epochs with a learning rate of 0.001 with Adam optimizer and with 0.8 learning rate decay. To ensure generalization, we train our network on indoor sequences and test on the outdoor_day1 sequence.

Table 4.4 shows the quantitative comparison with unsupervised-learning methods. Our model achieves the second best accuracy, following [180], and the best sharpness (FWL) among the existing methods. Notice that [180] was trained on the outdoor_day2 sequence, which is a similar driving sequence to the test one, while the other methods were trained on drone data [31]. Hence [180] might be overfitting to the driving data, while ours is not, by the choice of training data.

Additional qualitative results of our unsupervised learning setting are shown in Fig. 4.10. We compare our method with the state-of-the-art unsupervised learning [65]. Our results resemble the GT flow. See Tab. 4.4 for the quantitative result.

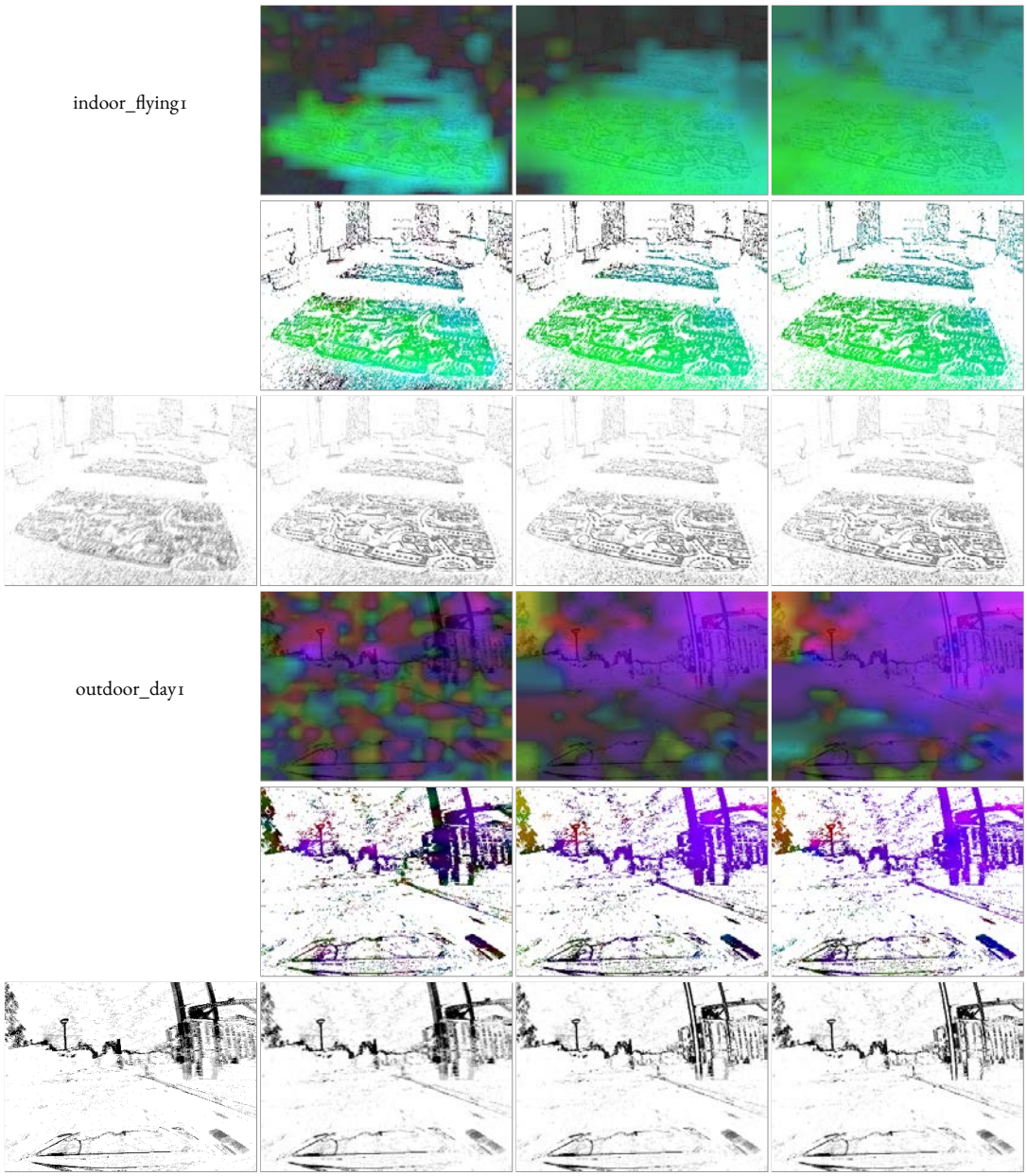
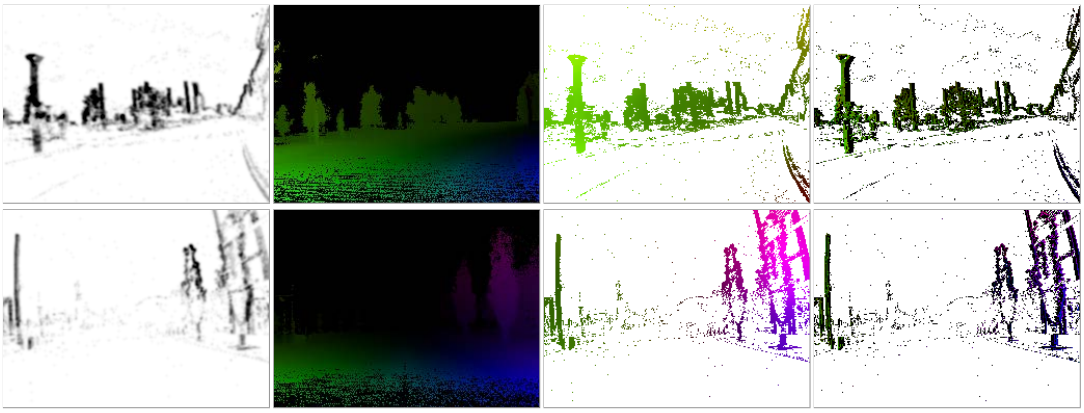


Figure 4.9: *Effect of the multi-scale approach.* For each sequence, the top row shows the estimated flow, the middle row shows the estimated flow masked by the events, and the bottom row shows the IWEs.

Table 4.4: Results of unsupervised learning on MVSEC's outdoor_day1 sequence.

	$dt = 1$			$dt = 4$		
	AEE ↓	%Out ↓	FWL ↑	AEE ↓	%Out ↓	FWL ↑
EV-FlowNet[180]	**0.32**	**0.00**	–	**1.30**	**9.70**	–
EV-FlowNet (retrained)[118]	0.92	5.40	–	–	–	–
ConvGRU-EV-FlowNet[65]	0.47	0.25	0.94	1.69	12.50	0.94
Our EV-FlowNet using (4.9)	0.36	0.09	**0.96**	1.49	11.72	**1.11**



(a) Input events (b) GT (c) Our EVFlowNet with (4.9) (d) USL[65]

Figure 4.10: *Result of our DNN on the MVSEC outdoor sequence.* Our DNN (EV-FlowNet architecture) trained with (4.9) produces better result than the state-of-the-art unsupervised learning method[65]. For a quantitative comparison, see Table 4.4.

4.3.8 Sensitivity Analysis

The choice of loss function.

Table 4.5 shows the results on the MVSEC benchmark for different loss functions. We compare the (squared) gradient magnitude, image variance, average timestamp[180], and normalized average timestamp[65]. The gradient magnitude and image variance losses produce the best accuracy compared with the two average timestamp losses. Quantitatively, the image variance loss gives competitive results with respect to the gradient magnitude. However, for the reasons described in Sec. 4.2.2, and because the image variance sometimes overfits, we use gradient magnitude. Both average timestamp losses are trapped in the global optima which pushes all events out of the image plane, hence, the provide very large errors (marked as "> 99" in Tab. 4.5). This effect is visualized in Fig. 4.11.

Table 4.5: Sensitivity analysis on the choice of loss function (MVSEC, $dt = 4$). The contrast and gradient magnitude functions provide notably better results than the losses based on average timestamps.

	indoor_flying1		indoor_flying2		indoor_flying3		outdoor_day1	
	AEE ↓	%Out ↓	AEE ↓	%Out ↓	AEE ↓	%Out ↓	AEE ↓	%Out ↓
Gradient magnitude[45]	**1.68**	12.79	2.49	26.31	2.06	18.93	**1.25**	**9.19**
Image variance[47]	1.70	**11.25**	**2.18**	**21.91**	**1.93**	**15.84**	1.82	15.89
Avg. timestamp[180]	>99	>99	>99	>99	>99	>99	>99	>99
Norm. avg. timestamp[65]	>99	>99	>99	>99	>99	>99	>99	>99

(a) Input events (b) Gradient Magnitude (c) Variance (d) Avg. timestamp[180] (e) Norm. avg. timestamp[65]

Figure 4.11: *IWEs for different loss functions.* Average timestamp losses overfit to undesired global optima, which pushes most events out of the image plane.

Remark: Maximization of (4.6) does not suffer from the problem mentioned in[65] that affects the average timestamp loss function, namely that the optimal flow warps all events outside the image so as to minimize the loss (undesired global optima shown in Fig. 4.11d-4.11e). If most events were warped outside of the image, then (4.6) would be smaller than the identity warp, which contradicts maximization.

The regularizer weight.

Table 4.6 shows the sensitivity analysis on the regularizer weight λ in (4.9). $\lambda = 0.0025$ provides the best accuracy in the outdoor sequence, while $\lambda = 0.025$ provides slightly better accuracy in the indoor sequences. Comparing their accuracy differences, we use the former because it has a higher accuracy gain.

4.3.9 Limitations

Like previous unsupervised works[180,65], our method is based on the brightness constancy assumption. Hence, it struggles to estimate flow from events that are not due to motion, such as those caused by flickering lights. SL and SSL methods may forego this assumption,

Table 4.6: Sensitivity analysis on the regularizer weight (MVSEC data, $dt = 4$).

	indoor_flying1		indoor_flying2		indoor_flying3		outdoor_day1	
	AEE ↓	%Out ↓	AEE ↓	%Out ↓	AEE ↓	%Out ↓	AEE ↓	%Out ↓
$\lambda = 0.0025$	1.68	12.79	2.49	26.31	2.06	18.93	1.25	9.19
$\lambda = 0.025$	1.52	9.07	2.39	26.26	1.94	18.44	1.86	17.11
$\lambda = 0.25$	1.89	16.54	3.19	36.95	2.91	30.85	2.57	27.86

but they require high quality supervisory signal, which is challenging due to the HDR and high speed of event cameras.

Like other optical flow methods, our approach can suffer from the aperture problem. The flow could still collapse (events may be warped to too few pixels) if tiles become smaller (higher DOFs), or without proper regularization or initialization. Optical flow is also difficult to estimate in regions with few events, such as homogeneous brightness regions and regions with small apparent motion. Regularization fills in the homogeneous regions, whereas recurrent connections (like in RNNs) could help with small apparent motion.

4.4 Conclusion

We have extended the CM framework to estimate dense optical flow, proposing principled solutions to overcome problems of overfitting, occlusions and convergence without performing event voxelization. The comprehensive experiments show that our method achieves the best accuracy among all methods in the MVSEC indoor benchmark, and among the unsupervised and model-based methods in the outdoor sequence. It is also competitive in the DSEC optical flow benchmark. Moreover, our method delivers the sharpest IWEs and exposes the limitations of the benchmark data. Finally, we show how our method can be ported to the unsupervised setting, producing remarkable results. We hope our work unlocks future optical flow research on stable and interpretable methods.

5

Event-by-event Optical Flow Estimation

5.1 Fast Correlation-based Flow Estimation

Event cameras[91,41] have led to rethinking visual processing for various computer vision tasks because their operating principle and output data are fundamentally different from those of conventional, frame-based cameras. These bio-inspired sensors naturally respond to the scene dynamics and offer advantages, such as low latency, high dynamic range (HDR) and data efficiency, which need to be unlocked with new algorithms[43]. Neuromorphic principles have been a major source of inspiration for such novel algorithms and hardware, especially in motion estimation tasks[64,16,65].

Event-based optical flow estimation methods can be broadly classified as *packet*-based or *event-by-event*–based depending on how events are processed and update the estimator's output. Packet-based methods process a batch of events (e.g., events in a fixed time window, say 10–100 ms, or a fixed number of events, typically 30k–1M), hence they require some waiting time before processing (inference) starts[46,10,146,96]. They trade off the high-speed advantages of event data for accuracy. Prior work has proposed adaptations of classical frame-based methods (block matching[96], Lucas-Kanade[11]), spatio-temporal plane-fitting[10,1], time-surface matching[109], and contrast-maximization methods[176,46,146]. While the above methods are model-based (optimization) methods, Artificial Neural Networks (ANN)[179,180,56,34,87] are also batch-based, and are inspired by frame-based ANN architectures[160,77], thus requiring data conversion into a tensor representation, such as voxel grids. ANNs achieve current

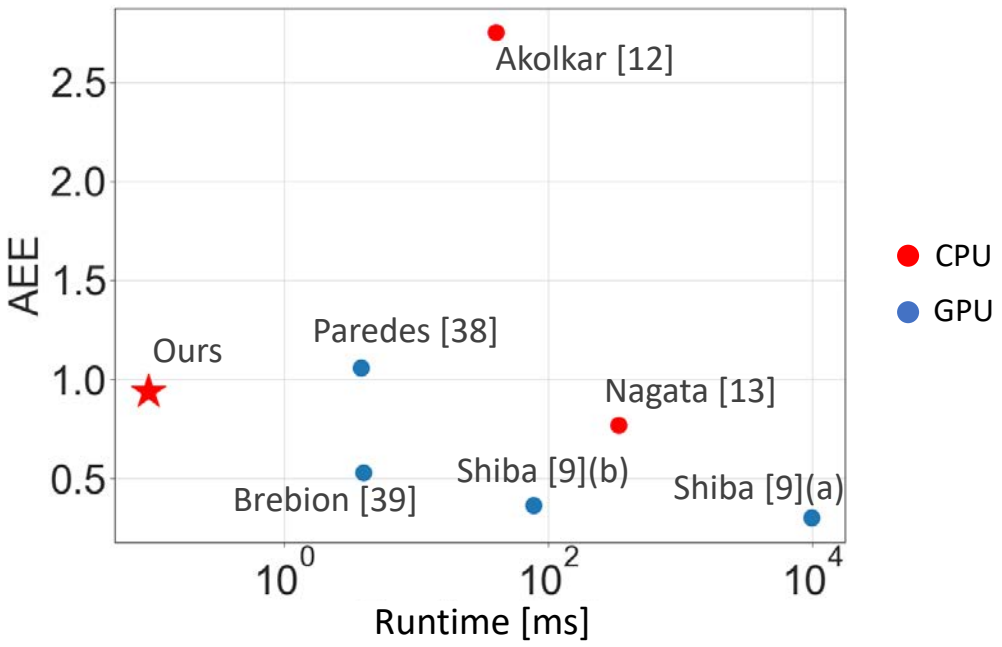


Figure 5.1: *Runtime vs. accuracy comparison* for various event-based optical flow estimation methods. Results are on outdoor data of the MVSEC benchmark[179] (see also Tab. 5.2). Accuracy is measured based on Average Endpoint Error (AEE). The numbers in the diagram indicate the reference numbers within[147].

state-of-the-art accuracy for optical flow estimation and high speed[56,146], but require power-hungry GPUs and lack interpretability.

On the other hand, event-by-event methods process every event incrementally as it occurs (without waiting time), aiming to leverage the camera's low-latency advantage[16,28]. Many event-by-event methods, such as Spiking Neural Networks (SNNs), are inspired by the brain (i.e., *neuromorphic*), since the neural circuits of visual processing are thought to be event-driven. While previous work propose SNN architectures[114,120,65], they comprise low-level physiological parameters of neurons (e.g., membrane potentials) that are difficult to interpret, validate and adjust to improve the estimation accuracy. Indeed, insects and mammals have different low-level underlying mechanisms, while they have similar algorithmic steps to transform light into motion[23]. Hence, it is important to find abstracted logical operations of motion estimation, rather than to mimic the entire physiological properties of neurons. From a practical point of view, most event-by-event methods have been tested on simple scenes, as opposed to the more complex real-world scenes and publicly-available benchmarks of batch-based methods[137]. This may be attributed to the use of tailored hardware[64], strong assumptions of the scene, limited problem settings[16] or the difficulty in defining event-by-

event benchmarks on real data with μs resolution. Hence, it is important to explore event-by-event motion estimation algorithms that can solve complex, real-world problems.

This work leverages insights from neuroscience, especially from the classical Barlow-Levick model[9], and proposes a novel optical flow estimation scheme based on triplet matching. In contrast to previous batch-based methods, it requires only three events for estimation, which opens the door to future real-time incremental motion estimation methods. Compared to previous event-by-event approaches, it is tested on publicly-available optical flow benchmarks to demonstrate its capability to handle real-world scenes with comparable results. Additionally, it is based on logical operations, which enables a simple and efficient data structure implementation and execution on standard CPUs. In summary, our contributions are twofold: (*i*) we present a novel event-by-event algorithm for optical flow estimation, theoretically derived from neuroscience insights, and (*ii*) we practically demonstrate that it achieves comparable results as prior work while only requiring a CPU and being faster than optimization-based algorithms (Fig. 5.1).

The signal processing in this work materializes the ideas in current neuroscience models, shedding light on what the strong and weak scenarios are, in order to improve the models.

5.2 Methodology

5.2.1 Event Camera

Event cameras acquire visual data in the form of asynchronous per-pixel brightness differences called "events"[91,43]. An event $e_k \doteq (t_k, \mathbf{x}_k, p_k)$ is triggered as soon as the logarithmic brightness at the pixel $\mathbf{x}_k \doteq (x_k, y_k)^\top$ exceeds a preset threshold. Here, t_k is the timestamp of the event with μs resolution, and polarity $p_k \in \{+1, -1\}$ is the sign of the brightness change (i.e., increase vs. decrease, respectively).

5.2.2 Triplet Matching

The idea of the triplet matching comes from neuroscience models by Hassenstein-Reichardt[68] and Barlow-Levick[9]. These correlator models estimate motion by computing pairwise neural activities (e.g., spikes) in space and time[23]. Especially,[42] suggests that triplet correlations (the product of pairwise correlations for three spikes in space-time) improve motion estimation accuracy. Here, we introduce the idea of the triplet-matching method as logical operations in space-time coordinates. We build an incremental (event-by-event) estimation algorithm, and extend it into batch mode for testing because benchmarks are specified on a batch basis.

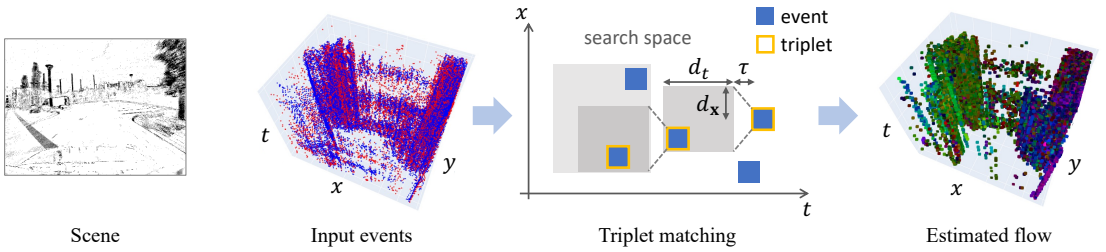


Figure 5.2: *Triplet matchig algorithm.* Triplet-matching algorithm seeks spatially and temporally neighboring events in an event-by-event manner, and provides event-based flow $\mathbf{f}_k$. Note this is an example of batch estimation given the input events.

Algorithm 1 Triplet matching algorithm

Input: $e_k, \mathcal{H}^{k-1}$
Output: $\mathcal{H}^k, \mathbf{f}_k$

1: Find event neighborhood H_k in (5.1).
2: **for** $i \in H_k$ **do**
3: Search for triplet candidates (5.2).
4: Collect triplet $T = (k, i, j)$
5: **end for**
6: Calculate $\mathbf{f}_k \leftarrow \{T\}$ in (5.3)
7: Update $\mathcal{H}^k \leftarrow \mathcal{H}^{k-1}, H_k$

Incremental Estimation

It consists of two main steps: *search* and *update* (Algorithm 1). Events are split by polarity, following the idea of ON- and OFF- circuits in the brain[23]. The search step finds triplets of events that are aligned (i.e., correlated) in space-time assuming a constant velocity model (Fig. 5.2). One of the events in the triplet is the incoming event, and the other two events are searched for within its space-time neighborhoods of size $d_{\mathbf{x}}, d_t$. The search has two steps: first the set of all potential 2nd events is determined; then the set of all potential 3rd events (compatible with the previous two in the triplet) refines the search. In the update step, every triplet of events is characterized by a different velocity. The velocity (flow) $\mathbf{f}_k$ for the incoming event e_k is computed as the average of the velocities of all triplets. Later, for benchmarking purposes, the flow is voxelized (quantized on a space-time grid) and smoothed.

In the *search step*, since event data are sorted by timestamp t, we use index maps to make the search efficient, with complexity $O(N_e \log N_e)$. The index map H_k of an event e_k consists

of the indices of its space-time neighbors:

$$H_k = \{i \mid t_k - \tau - d_t \leq t_i \leq t_k - \tau \text{ and } \|\mathbf{x}_k - \mathbf{x}_i\| \leq d_\mathbf{x}\}. \tag{5.1}$$

Parameters d_t and $d_\mathbf{x}$ decide the maximum admissible velocity of the flow, and τ is a *refractory period*, which limits the search space by assuming neighboring events in the moving edge do not exist at the same timestamp. d_t can also be interpreted as the *delay* in the Barlow-Levick model. For each new event e_k, we build a set of index maps $\mathcal{H}^k = \{H_i\}_{i=1}^k$ and output a set of event triplets $\{T\} \doteq \{(e_k, e_i, e_j)\}$. To find the triplet match we look for event indices j that have roughly constant velocity with the event pairs (e_k, e_i) where $i \in H_k$:

$$J_{k,i} = \{j \in H_i \mid t_i - \tau - d_t \leq t_j \leq t_i - \tau \text{ and } \mathbf{x}_i - \mathbf{x}_j = \mathbf{x}_k - \mathbf{x}_i\}. \tag{5.2}$$

The *update step* calculates the flow $\mathbf{f}_k$ and updates the index map $\mathcal{H}^k$. $\mathcal{H}^k$ is obtained by adding new H_k to $\mathcal{H}^{k-1}$ and removing old index maps (we keep the latest 20000 index maps per polarity). The flow $\mathbf{f}_k$ is obtained as the weighted average

$$\mathbf{f}_k \doteq \frac{\sum_T w_T \mathbf{v}_T}{\sum_T w_T}, \tag{5.3}$$

where $\mathbf{v}_T \doteq (\mathbf{x}_j - \mathbf{x}_k)/(t_j - t_k)$ is the velocity of each triplet. Since (5.3) gives accurate flow if the triplet is caused by the same scene edge, we use the weight w_T to estimate the probability that the triplet belongs to the same edge. Assuming constant velocity, if e_j is produced by the same edge that generates e_k and e_i, the expected timestamp of e_j is given by $\hat{t}_j = t_i - \delta$, where $\delta \doteq t_k - t_i$. Therefore, to account for errors in the timestamps between $\hat{t}_j$ and t_j, we set the weight $w_T \doteq \mathcal{N}(t_j; \hat{t}_j, \delta^2)$, where $\mathcal{N}$ is the Gaussian density function.

Batch Estimation

We extend the incremental (event-by-event) estimator to batch mode because current benchmarks are batch-based. For a set of events $\mathcal{E} \doteq \{e_k\}_{k=1}^{N_e}$, we create the index maps $\mathcal{H}^{N_e}$ first, which takes $O(N_e^2 \log N_e)$. Then the flow is calculated looping over each event using Algorithm 1. The overall computational complexity is $O(N_e^2 \log N_e)$.

For benchmarking with ground truth, the event-wise flow is converted into a voxel-wise flow, which also enhances space-time coherence. We quantize the time coordinates of $\mathbf{f}_k$ into bins, and take the average of the $\{\mathbf{f}_k\}$ that lie in each voxel. We also apply a non-zero average filter (take average of only non-zero values) with kernel size 3×3 for spatial smoothing.

The computational complexity of both approaches is summarized in Tab. 5.1. For comparison, we also report those of optimization-based methods: Contrast Maximization (CMax)[46]

and time-surface matching[109]. The latter methods require additional complexity for the number of iterations N_{iter}, which is inefficient. We report runtime comparisons in Sec. 5.3.3.

Algorithm 2 Triplet matching: Batch

Input: $\mathcal{E}$
Output: $\mathbf{f}, \mathcal{H}^{N_e}$
1: **for** $k = 1$ to N_e **do**
2: Search $H_i = \{i \mid t_k - \tau - d_t \leq t_i \leq t_k - \tau \text{ and } \|\mathbf{x}_k - \mathbf{x}_i\| \leq d_{\mathbf{x}}\}$
3: **end for**
4: **for** $k = 2$ to N_e **do**
5: Calculate $\mathbf{f}_k, \mathcal{H}^k$ using Algorithm 1 with e_k, cH^{k-1}
6: **end for**
7: **return** $\mathbf{f}, \mathcal{H}^k$

Table 5.1: Complexity of algorithms, for batch estimation and event-by-event estimation.

	Batch	Event-by-event
CMax[46]	$O(N_{\text{iter}}(N_e + N_p))$	–
Nagata et al.[109]	$O(N_{\text{iter}}(N_e + N_p))$	–
Ours	$O(N_e^2 \log N_e)$	$O(N_e \log N_e)$

5.3 Experiments

5.3.1 Datasets and Evaluation Metrics

The MVSEC dataset[178] is a standard dataset for optical flow estimation[180,56,65,146]. The data consists of event camera, LiDAR, and camera poses. The event camera (mDAVIS346 camera[158]) provides events, grayscale frames and IMU data (346 × 260 pix). The ground truth optical flow is provided as the motion field from the camera velocity and the depth of the scene[179]. The sequences are indoors with a drone and outdoors with a car, and we evaluate on 63.5 million events spanning 265 seconds from both outdoor and indoor sequences.

We measure optical flow accuracy to evaluate our method. The metrics are the Average Endpoint Error (AEE) and the percentage of pixels with AEE greater than 3 pixels (% Out). The time intervals for evaluation are $\Delta t = 1$ grayscale frame (at ≈ 45Hz, i.e., 22.2ms) and $\Delta t = 4$ frames (89ms). Flow accuracy is evaluated only in pixels with valid ground truth. All experiments use $d_{\mathbf{x}} = \sqrt{2}$ pix, $d_t = 100$ms and $\tau = 3$ms.

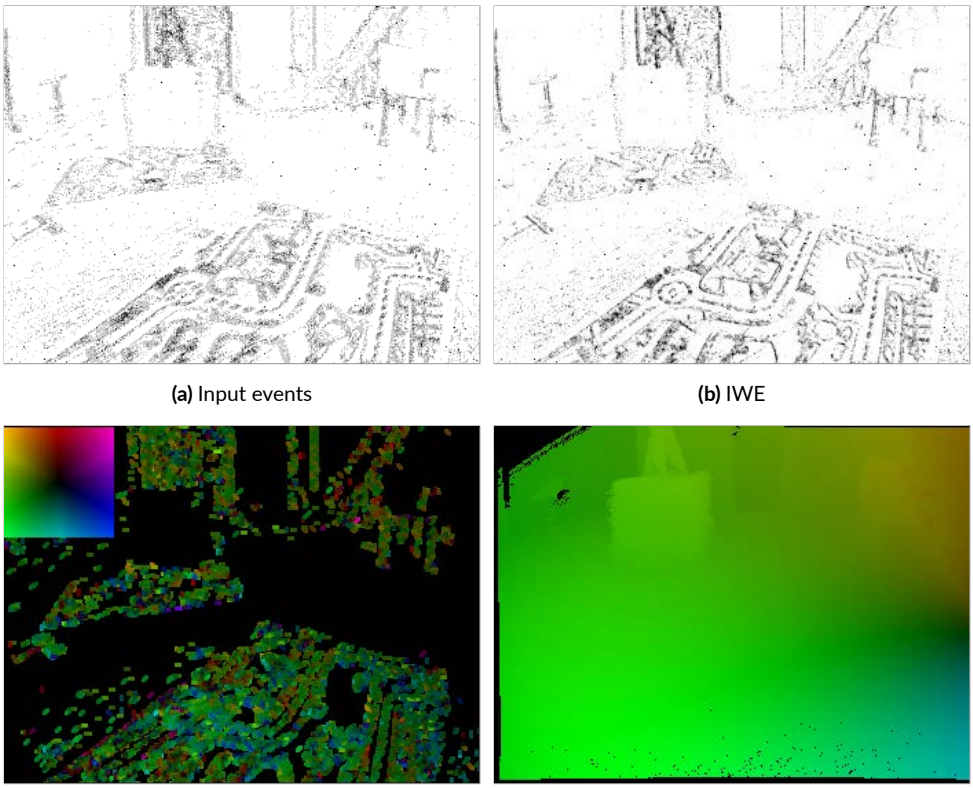
Table 5.2: Results on MVSEC dataset[179]. Methods are presented as unsupervised learning-based (USL) or model-based (MB). For brevity, EV-FlowNet is abbreviated as EVFN. Nagata et al.[109] evaluate on shorter time intervals; for comparison, we scale the errors to $\Delta t = 1$.

		outdoor_day1		indoor_flying1		indoor_flying2	
	$\Delta t = 1$	AEE ↓	%Out ↓	AEE ↓	%Out ↓	AEE ↓	%Out ↓
USL	EVFN[180]	0.32	0.00	0.58	0.00	1.02	4.00
USL	EVFN (retrain)[118]	0.92	5.40	0.79	1.20	1.40	10.90
USL	FireFlowNet[118]	1.06	6.60	0.97	2.60	1.67	15.30
USL	ConvGRU-EVFN[65]	0.47	0.25	0.60	0.51	1.17	8.06
USL	MultiCM-EVFN[146]	0.36	0.09	–	–	–	–
MB	Nagata et al.[109]	0.77	–	0.62	–	0.93	–
MB	Akolkar et al.[1]	2.75	–	1.52	–	1.59	–
MB	Brebion et al.[15]	0.53	0.20	0.52	0.10	0.98	5.50
MB	MultiCM[146]	0.30	0.10	0.42	0.10	0.60	0.59
MB	Ours	0.94	3.08	1.05	2.90	1.68	13.44
	$\Delta t = 4$						
USL	EVFN[180]	1.30	9.70	2.18	24.20	3.85	46.80
USL	ConvGRU-EVFN[65]	1.69	12.50	2.16	21.51	3.90	40.72
USL	MultiCM-EVFN[146]	1.49	11.72	–	–	–	–
MB	MultiCM[146]	1.25	9.21	1.69	12.95	2.49	26.35
MB	Ours	3.60	49.04	4.06	53.88	6.39	71.82

We also show qualitative results on the DSEC dataset[55] and the ECD dataset[105]. Both datasets are widely used for motion estimation[47,177,63,103,146]. Each sequence of the DSEC dataset consists of events from Prophesee Gen3 event cameras (640×480 pixels) and ground truth optical flow (at 10 Hz) with the scene depth from a LiDAR. Each sequence of the ECD dataset provides events, frames, calibration, and IMU data (at 1 kHz) from a DAVIS240C (240×180 pix)[13], as well as ground truth camera poses (at 0.2 kHz).

5.3.2 Optical Flow Estimation Accuracy

Table 5.2 comprises flow estimation accuracy results on the MVSEC benchmark. The top part of the table reports results for $\Delta t = 1$, and the bottom part reports $\Delta t = 4$. The methods in the table are categorized as unsupervised learning-based (USL), i.e., using a Deep Neural



(a) Input events **(b)** IWE

(c) Estimated flow **(d)** Ground truth flow

Figure 5.3: *Optical flow results on MVSEC data.*

Network (DNN) on grid-converted events, and model-based (MB). Results for $\Delta t = 1$ are thorough, with our method in the middle accuracy range among all methods. Results for $\Delta t = 4$ are not as complete because the literature does not report them (especially most model-based methods). While a thorough comparison for $\Delta t = 4$ is difficult, our error is roughly four times bigger than for $\Delta t = 1$, which makes sense, and it is consistently 2.5–3 times bigger than that of the most accurate method[146] for both $\Delta t = \{1, 4\}$. The fact that for longer time intervals batch-based methods ($\Delta t = 4$) achieve higher accuracy than our method may be attributed to the fact that our method is event-by-event, so it does not leverage long-term temporal smoothing, which would improve robustness to noise.

Figure 5.3 shows qualitative results. As it is noticeable, the events displaced using the estimated flow produce sharp images of warped events (IWEs[46]). The Flow Warp Loss[155] measures the sharpness of the IWE qualitatively: 1.154 for outdoor_day1, 1.157 for indoor_flying1, and 1.248 for indoor_flying2, where FWL larger than 1 indicates sharper than the identity warp baseline (i.e., zero flow). The figure also shows the estimated flow; notice that our

method produces a flow vector for each event (Fig. 5.2), whereas it is common to display the flow for every pixel (image-based legacy). Hence, Fig. 5.3 shows a 2D collapsed version of the estimated space-time optical flow field, for visual comparison with the ground truth. The flow is most reliably estimated in regions where events happen, i.e., scene edges. Further spatial and temporal smoothness could be enhanced if needed: for example, homogeneous brightness regions between edges could be filled in by some prior, such as a regularizer or in-painting algorithm.

5.3.3 Runtime Comparison

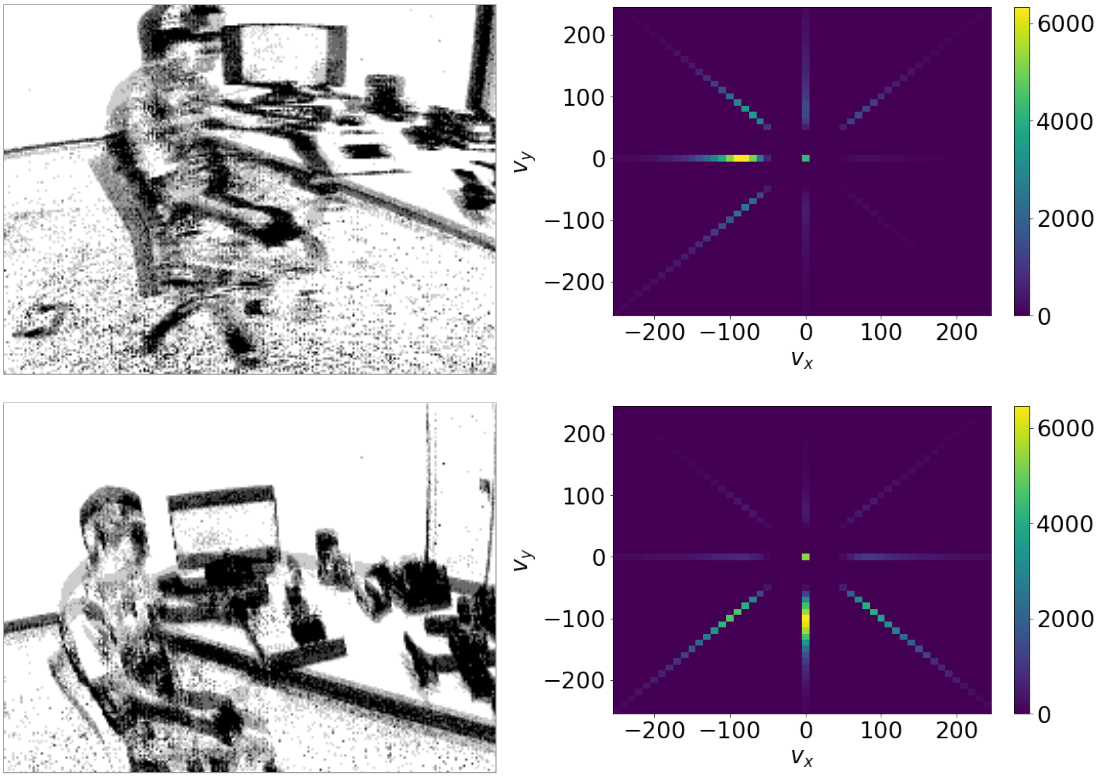
The proposed method runs in an event-by-event manner and hence trades off accuracy for speed, compared with batch-based methods. We showed computational complexity comparison in Tab. 5.1. Now, we conduct the runtime comparison among several previous work. We use Python (3.9.12) on CPUs (Mac M1 2020, 8 Cores). We process 300k events incrementally and average the resulting runtimes. The results are shown in Fig. 5.1. Our method achieves the fastest runtime among compared methods: 0.0934 milliseconds (>10 kHz). Note that many methods in the literature, such as the second[118] and third[15] fastest ones, use GPUs, while ours can run natively on CPUs. These fast and lightweight characteristics of the proposed method are important for future robotics applications of event cameras on resource-constrained platforms.

5.3.4 Effect of Pixel Quantization

A limitation of the proposed method is the quantization of the flow direction since the search for the second event in the triplet is limited to the 8 neighboring pixels of the current event. To illustrate it, we conduct experiments on the *dynamic_translation* sequence from the ECD dataset[105]. Figure 5.4 shows the distribution of $\mathbf{v}_T$ over all events (assuming a planar translation model, i.e., constant velocity over all pixels). Similar to the SNN proposed in[114], $\mathbf{v}_T$ is constrained to eight cardinal directions. However, in contrast to[114], which quantizes both the direction and magnitude of the flow, our method can estimate a continuum of magnitudes. The distributions are spread around a main direction and its two neighboring ones, which is due to the small aperture (5×5 pix) used for each triplet.

5.3.5 Results on Higher Spatial Resolution

For completeness, we also show qualitative results on DSEC, with a higher-resolution (640×480 pixels) event camera in Fig. 5.5. Similarly to Fig. 5.3, the estimated flow is shown as a 2D collapsed version of the estimated space-time optical flow field. The estimated flow (Fig. 5.5c) resembles the ground truth and provides a sharp IWE (Fig. 5.5b). These results demonstrate



(a) Input events

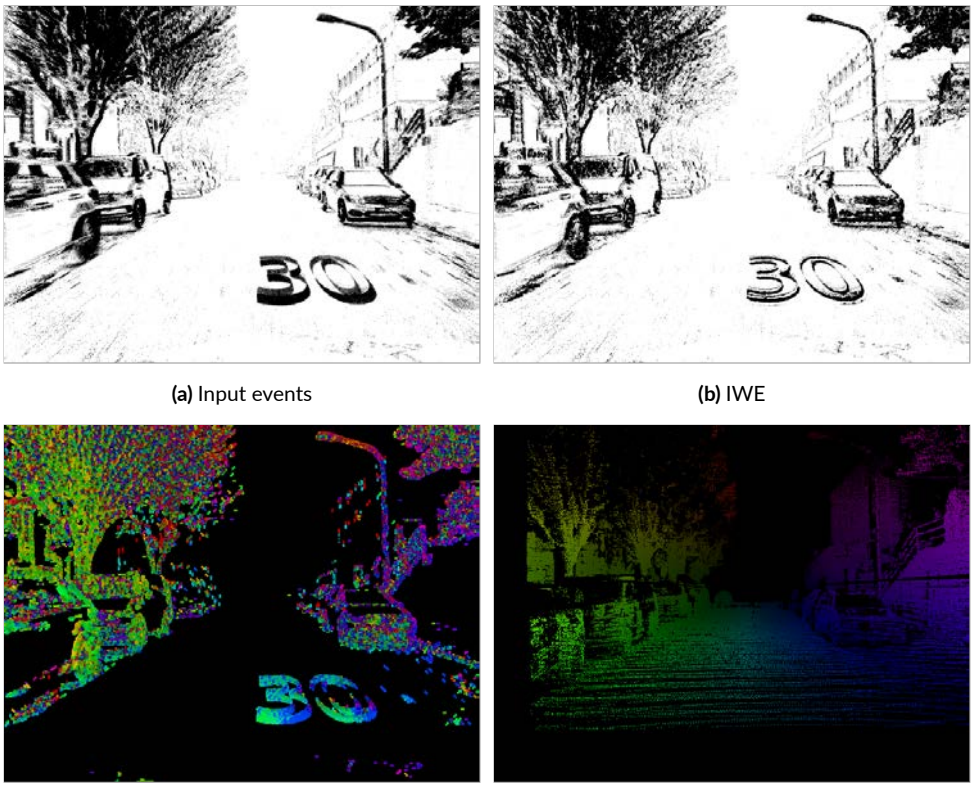
(b) Estimated velocities $\mathbf{v}_T$

Figure 5.4: *Effect of pixel quantization on ECD data.* In the top row the motion is dominantly horizontal, whereas in the bottom row it is vertical, as can be seen by the thickness of the edges (left) and the velocity distributions (right).

that the proposed method works on a recent high-resolution event camera. In this example, the same parameters (e.g., $d_{\mathbf{x}}$ in (5.1)) generalize to cameras with different resolutions, which have different sizes of the receptive field (space in the visual space) per pixel. However, due to the smaller pixel size, it sometimes suffers from the aperture problem (e.g., the "30" on the road). One might need to consider increasing the recursive search (5.2), such as a quadruple (the pairs of four events) and so on.

5.4 Conclusion

We proposed a novel event-based optical flow estimation scheme based on triplet matching that runs in an event-by-event manner. The proposed method was biologically plausible for event-based optical flow since it leverages knowledge from neuroscience. The experiments demonstrated that it is considerably fast ($>$ 10 kHz) on standard CPUs while providing com-



(a) Input events **(b)** IWE

(c) Estimated flow **(d)** Ground truth flow

Figure 5.5: *Optical flow results on DSEC data.*

parable results as prior batch-based algorithms. We hope that our work opens the door to real-time, realistic, incremental motion estimation methods and event-camera applications on resource-constrained devices.

6

Estimating Motion of Air Convection

6.1 Introduction

Sensing the flow of transparent media, such as air or water, is important for various applications from aerodynamics to gas leakage detection. Optical imaging is a useful tool to examine such transparent media because it can capture the media with high detail in space-time remotely. Among existing methods, schlieren imaging is a simple but efficient optical tool for seeing the "invisible" [142,143]: inhomogeneities in transparent media that are not necessarily perceived by the naked eye. It requires simple recording settings: lenses, cameras, and mirrors or background patterns to image how light rays deviate due to refractive index variations in the media. While it was initially conceived as a visualization technique, recent developments in schlieren and shadowgraphy fields have extended the usage to velocimetry [143,144]. However, it requires a high-speed camera with a large spatial resolution to analyze the velocity of the flow, such as convection. This is not only a constraint for real-world applications but also a limitation of the methodology because: (*i*) achieving high shutter speeds requires unnaturally bright illumination, which is not always practical, (*ii*) transmitting and processing the large amount of redundant data acquired involves high bandwidth, storage, and power-hungry components, and (*iii*) regardless of the large power consumption, the trade-off between speed and spatial resolution limits accuracy in estimating the flow velocity.

Event cameras [91,41] are novel bio-inspired sensors that respond to pixel-wise intensity changes, which are not always visible to conventional frame-based cameras. They offer advantages

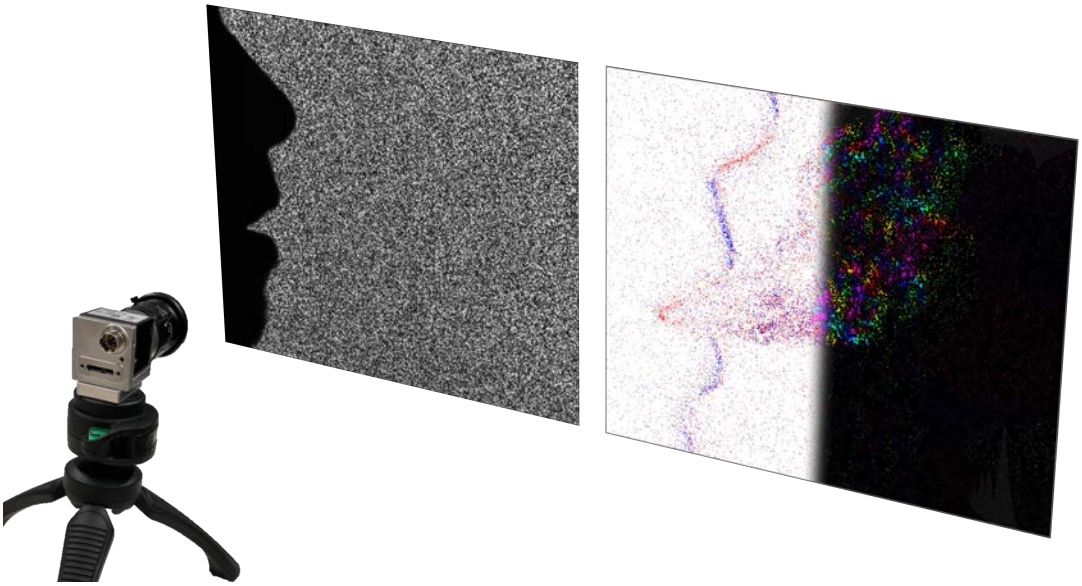


Figure 6.1: In background-oriented schlieren imaging local density gradient variations between a camera and a background pattern lead to tiny perceived changes on the image plane. We show how to combine events and frames to calculate optical flow of the schlieren scene and how to leverage the advantages of event cameras to visualize gas streams such as the human breath.

such as high speed, high dynamic range (HDR), low power consumption, and data efficiency (temporal redundancy suppression)[43], which makes them potential candidates to overcome the limitations of traditional (i.e., frame-based) schlieren techniques. However, despite these potential capabilities, the application of event cameras to imaging applications is yet to be explored and developed.

This chapter presents a novel technique, event-based background-oriented schlieren (BOS), for sensing air convection with event cameras and proposes a novel method to estimate the temporal derivative of air density from events and frames (Fig. 6.1). Throughout the chapter, we tackle the following challenges of event-based BOS: (*i*) Theory. There is no established mathematical theory for event-based schlieren techniques. (*ii*) Data. Event cameras sense only increments of schlieren as opposed to the larger differences with respect to a reference in frame-based BOS. (*iii*) Methodology. The origin of events in BOS (flickering because they happen only at the edges of the background pattern) and large amounts of noise are novel and difficult for previous work in event-based vision. (*iv*) Evaluation. The true ground truth of the air density is not easy to obtain, hence we need some proxy ground truth and baselines.

First, we develop a theoretical connection between the schlieren and events, showing that event cameras can sense the inhomogeneities of transparent media in a more direct way (as flickering events) compared to frame-based cameras. Such direct sensing of schlieren through

event data enables us to observe air convection at high speed more precisely and under challenging lighting conditions. Second, we propose a novel method that extends the linearized event generation model with physically-inspired parameterization to estimate the temporal density fluctuation due to the schlieren. Third, to evaluate the estimated density change, whose real-world ground truth is not easy to obtain, we establish the evaluation method using optical flow, by revealing the theoretical connection between the temporal density change and optical flow (i.e., pixel displacement). Using a co-located frame-based camera enables us to benchmark different methods of estimating temporal density change as a computer vision problem. The experimental results show that: (*i*) our proposed method recovers the flow that corresponds to the temporal change of density gradient by comparing with the standard frame-based methods and other baseline methods, (*ii*) flickering-like events are a more direct measurement of such schlieren, (*iii*) event cameras record the density inhomogeneities even in poor lighting conditions, which state-of-the-art frame-based algorithms cannot provide, and (*iv*) the high temporal resolution of event cameras enables slow-motion schlieren analysis.

The main technical contributions of this chapter are:

- A novel method for computation of schlieren combining events and frames (Secs. 6.3 and 6.4). The proposed method is rigorously obtained and well connected with the physical model of the sensors involved via the linearized event generation model.
- The first schlieren event-frames dataset (Sec. 6.5). We publicly provide recordings of several schlieren scenes by means of events and frames, at high resolution (1 Mpixel), accurately synchronized and calibrated using an in-house acquisition system.
- A thorough comparison with baseline methods despite the lack of truly ground truth data in this type of turbulent fluid dynamics phenomena (Sec. 6.6).

To the best of our knowledge, this is the first work showing the potential advantages of event cameras for schlieren imaging applications.

6.2 Related Work

6.2.1 Background-Oriented Schlieren

Schlieren photography was invented in 1864 to study the flow of air around objects moving at supersonic speed [142]. In contrast to other imaging and velocimetry techniques such as particle image velocimetry [127,33], it does not require any particle seeding in the media of interest. Among different schlieren-imaging techniques (see Tab. 6.1), BOS is a relatively recent technique since it utilizes digital image processing [133]. In BOS (Fig. 6.2), an object of interest

Table 6.1: Comparison of various schlieren imaging techniques and the physical quantities they measure.

Method	Observation
Shadowgraphy [73,126]	$\frac{\partial^2 \rho}{\partial \mathbf{x}^2}$
Toepler's schlieren photography [84,126]	$\frac{\partial \rho}{\partial \mathbf{x}}$
Laser speckle photography [126]	$\frac{\partial \rho}{\partial \mathbf{x}}$
Frame-based BOS [126]	$\frac{\partial \rho}{\partial \mathbf{x}}$
Event-based BOS (this work)	$\frac{\partial^2 \rho}{\partial t \partial \mathbf{x}}$

with density variations (e.g., the hot air stream from a burning candle) is placed between the camera and a constant (non-moving) background pattern. The schlieren generates complex deformation to the background pattern, which is observed by cameras as the apparent motion of the background pattern with respect to a reference image (without density variations) [133]. Different methods have been proposed to compute the displacement vector field of the apparent motion, such as using cross-correlation [60], optical flow [4], or wavelet-based analysis [139]. As equally important as the data processing method is the data acquisition setup. Best practices for parameter settings, such as the distance from the camera to the background and the media, are provided in [143].

BOS has been used to image various transparent media, such as shock waves from explosions [150], turbulent flows [144], and shock waves underwater [69]. Also, the background pattern of BOS can be extended to natural images [67], which allows us to image the flow with large field-of-view (FOV). In [70], BOS is utilized to visualize supersonic jets in flight, by leveraging the natural vegetation of the terrain seen from above as the constant background pattern. The large FOV is one of the unique characteristics of BOS unlike other schlieren techniques, which enables measuring natural outdoor scenes [143]. Notwithstanding, BOS can be used as input to other analysis tools, such as Dynamic Mode Decomposition (DMD) to reveal the main frequency modes of variation of the signal in space and time [163], which ultimately inform about the physical parameters of the turbulent flow. Recently, some works have extended BOS from an imaging technique to a quantitative method, e.g., [165] measures the density of axisymmetric supersonic flow. In [144], a method is proposed to extract velocity data from flows. For this application, Kymography works better than classical image correlation, and the self-similarity of round turbulent jet velocity appears in the schlieren results.

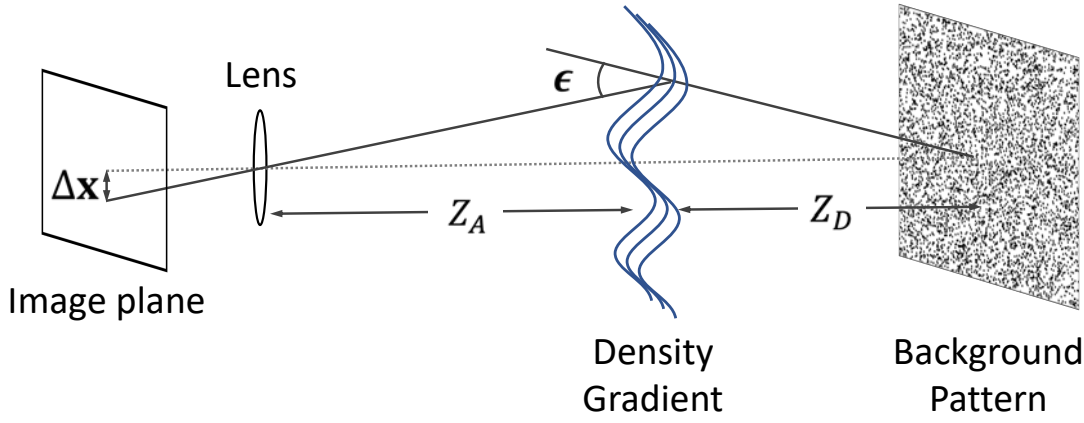


Figure 6.2: Background-Oriented Schlieren (BOS) setup.

6.2.2 Event Cameras

Event cameras are a relatively new technology compared to BOS imaging with standard frame-based cameras. Since the 2008 seminal work[91], they have been slowly commercialized and explored in computer vision and robotics for various applications. Event cameras naturally respond to motion in the scene at high speed and HDR in a data-efficient manner, hence large progress has been made in motion-related tasks, such as optical flow estimation[10,179,65,146], ego-motion estimation[46,113,121,148], SLAM[81,129,177,71], or video deblurring and frame interpolation[172,164,50].

Only recently, the larger spatial resolution of event cameras and higher fill factor of their pixels[41,156] has enabled fine-detail applications that were not possible with older models. Some works have explored event cameras for detecting small changes in the scene. These include vibration monitoring[123], particle-image velocimetry[169], and time-resolved 3D fluid flow reconstruction via collimated illumination[167]. These works open another stack of event camera applications in the field of fluid dynamics. Event-based BOS aims at pushing the limits, by imaging and quantifying flow fields without any particle seeding.

6.3 Event-based Schlieren

6.3.1 Principles of Frame-based BOS

In frame-based BOS the schlieren object S (e.g., a gas with varying density) produces an apparent displacement of the background pattern, which is measured with respect to the initial state (i.e., the image acquired in the absence of density gradient). The displacement $\Delta\mathbf{x} \doteq (\Delta x, \Delta y)^\top$ is directly related to the small deflection angle $\varepsilon \doteq (\varepsilon_x, \varepsilon_y)^\top$ (Fig. 6.2) via

the distance from lens to S (Z_A), the distance from S to the background (Z_D), and the focal length of the lens f[126]:

$$\Delta\mathbf{x} \approx f\left(\frac{Z_D}{Z_D + Z_A - f}\right)\varepsilon. \tag{6.1}$$

On the other hand, for the refractive index n, the angle ε is the result of aggregating the spatial gradient $\partial n/\partial\mathbf{x}$ along the length Z of the schlieren object S on the optical axis:

$$\varepsilon = \frac{1}{n}\int \frac{\partial n}{\partial \mathbf{x}} dz = \frac{Z}{n_\infty}\frac{\partial n}{\partial \mathbf{x}}, \tag{6.2}$$

where the ambient-air refractive index is given as n_∞. Finally, n is related to the density ρ of the gas (schlieren object) via the Gladstone-Dale relation, $n = G\rho + 1$, with constant $G = 2.23 \times 10^{-4}\mathrm{m}^3/\mathrm{kg}$[126].

In short, the spatial gradient of the density $\partial\rho/\partial\mathbf{x}$ within a gas causing schlieren can be directly quantified by measuring the pixel displacement $\Delta\mathbf{x}$:

$$\Delta\mathbf{x} \propto \frac{\partial\rho}{\partial\mathbf{x}}, \tag{6.3}$$

as summarized in Tab. 6.1. Here, the displacement is measured against the initial state (the background pattern), hence the corresponding density-gradient field is the change with respect to the initial (also called "reference") state.

6.3.2 Principles of Event-based BOS

One of the main differences between frame-based BOS and event-based BOS is that event cameras only sense temporal changes in the scene, while the former measures the displacement between a reference frame and the current frame (Fig. 6.3). Hence, the key challenge is how we can relate events (the asynchronous intensity changes between two timestamps t_1 and t_2) to the density ρ. Since events are very noisy[62,43], accumulating the differences between far away timestamps to estimate the same displacement as frame-based BOS (6.3) leads to high noise levels[138,14], which makes it difficult to estimate this displacement with events.

In order to establish the theoretical connection between schlieren and events, let us first extend the previous frame-based BOS theory to compute the displacement between two *nearby* timestamps. Given frames at timestamps t_1, t_2, their displacements from a reference frame at t_{ref} (6.1) are $\Delta\mathbf{x}(t_{\text{ref}}, t_1)$ and $\Delta\mathbf{x}(t_{\text{ref}}, t_2)$. The optical flow $\mathbf{v}(\mathbf{x}) = \partial\mathbf{x}/\partial t$ between consecutive frames for small $\Delta t = t_2 - t_1$ is

$$\mathbf{v}(\mathbf{x}) = \frac{\Delta\mathbf{x}(t_{\text{ref}}, t_2) - \Delta\mathbf{x}(t_{\text{ref}}, t_1)}{\Delta t}. \tag{6.4}$$

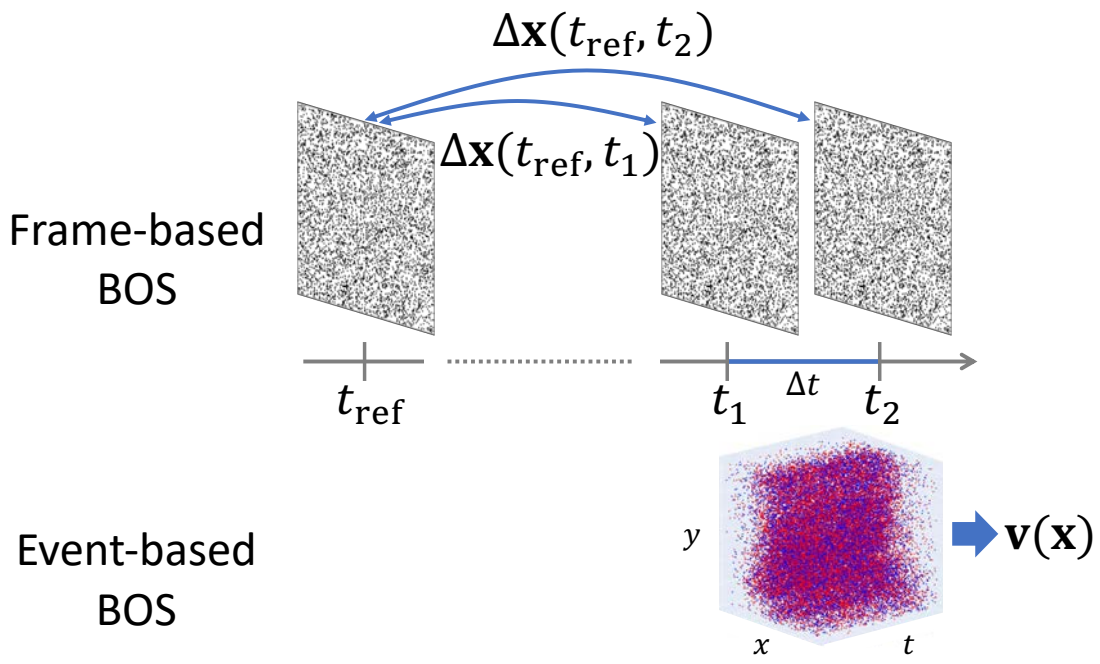


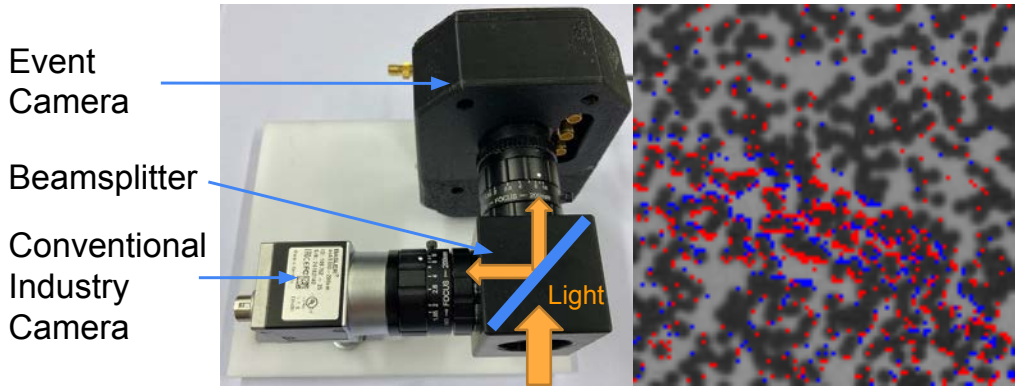
Figure 6.3: Frame-based BOS and event-based BOS.

From the frame-based BOS theory, the displacement at each timestamp can be related to the density gradient as follows (6.3):

$$\begin{aligned} \Delta\mathbf{x}(t_{\text{ref}}, t_1) &\propto \frac{\partial \rho_{t1}}{\partial \mathbf{x}}, \\ \Delta\mathbf{x}(t_{\text{ref}}, t_2) &\propto \frac{\partial \rho_{t2}}{\partial \mathbf{x}}. \end{aligned} \tag{6.5}$$

Plugging (6.5) into (6.4), using finite-difference approximations and Schwarz's theorem, gives:

$$\begin{aligned} \mathbf{v}(\mathbf{x}) &\propto \frac{1}{\Delta t}\left(\frac{\partial \rho_{t2}}{\partial \mathbf{x}} - \frac{\partial \rho_{t1}}{\partial \mathbf{x}}\right) \\ &= \frac{1}{\Delta t}\frac{\partial}{\partial \mathbf{x}}\left(\rho_{t2} - \rho_{t1}\right) \\ &\approx \frac{\partial}{\partial \mathbf{x}}\frac{\partial}{\partial t}\rho, \\ &= \frac{\partial}{\partial t}\frac{\partial}{\partial \mathbf{x}}\rho. \quad \text{(Schwarz's thm)} \end{aligned} \tag{6.6}$$



(a) Recording system **(b)** Events and frame

Figure 6.4: (a) Actual synchronized data recording system, combining an event camera and a frame-based camera via a beamsplitter (Sec. 6.5.1). (b) Data: events (red and blue, colored according to polarity) during a short time window overlaid on a grayscale frame (a 100×100 pixel region for better visualization).

That is, the optical flow between two nearby timestamps is related to the *temporal derivative of the density gradient* (see the last row of Tab. 6.1). Since events are the measurements between such nearby timestamps, the key question is how can optical flow (i.e., spatio-temporal derivative of the density) be estimated from event data.

6.4 Estimation Method

One of the main challenges of event-based BOS is its data modality: events generated by schlieren objects are sparse, happening only at the edges of the background pattern (Fig. 6.4b) and in a flickering form. Previous event-based optical flow estimation methods [146,147,179,65] often assume a *continuous, non-flickering apparent motion* of the visual patterns on the image plane (e.g., Chapters 4 and 5). Also, events triggered during the short time interval needed to capture fine details of the complex motion patterns are few compared to those in scenes from typical optical flow benchmarks [178,55]. Consequently, prior methods fail to produce accurate flow since they are not tailored to the schlieren scenario, as we show in Sec. 6.6.2. Due to these challenges, we propose a method that combines events and knowledge of the background pattern (e.g., frames) to estimate the flow. The proposed method extends the linearized event generation model (LEGM) [54,17,116,71], which has been used for modeling as complex motion as a rigid-body motion. In this work, we extend the LEGM towards further complex motion: optical flow. The overall pipeline is described in Fig. 6.5.

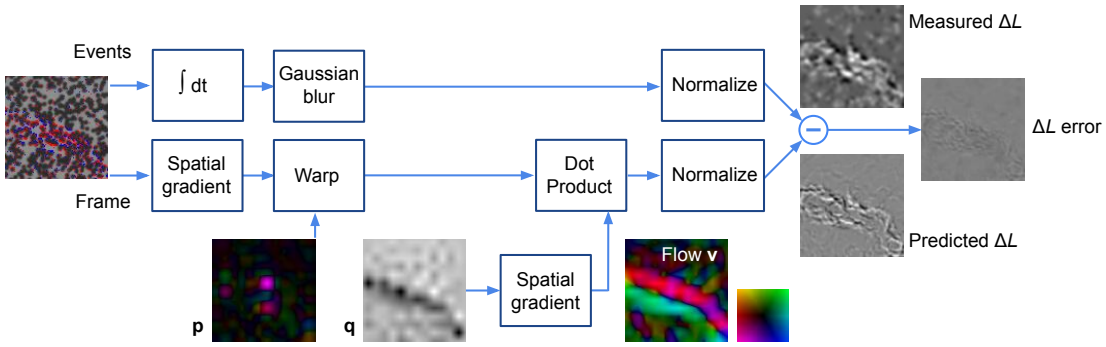


Figure 6.5: Block diagram of the objective E_{data} in (6.11). On the top branch, events are integrated in time using (6.8) and smoothed with a Gaussian kernel ($\sigma = 2$ pix) to produce the measured brightness increment image ΔL. The bottom branch shows how to compute the predicted brightness increment $\Delta\hat{L}$ from the frame and the unknowns of the problem: the translation field $\mathbf{p}$ and the Poisson parameters of the flow, $\mathbf{q}$. The optical flow $\mathbf{v}$ and $\mathbf{p}$ are pseudo-colored (color wheel is included). Same data as Fig. 6.4.

6.4.1 Event Generation Model

An event $e_k \doteq (\mathbf{x}_k, t_k, p_k)$ conveys that the logarithmic brightness L at pixel $\mathbf{x}_k$ changes by a specified contrast sensitivity C[91,43]:

$$\Delta L(\mathbf{x}_k, t_k) \doteq L(\mathbf{x}_k, t_k) - L(\mathbf{x}_k, t_k - \Delta t_k) = p_k\, C, \tag{6.7}$$

where polarity $p_k \in \{+1, -1\}$ is the sign of the brightness change, and Δt_k is the time since the last event at pixel $\mathbf{x}_k$. Given a set of events $\mathcal{E} \doteq \{e_k\}_{k=1}^{N_e}$, summing their polarities pixel-wise produces a brightness increment image:

$$\Delta L(\mathbf{x}) = \sum_k p_k C\, \delta(\mathbf{x} - \mathbf{x}_k), \tag{6.8}$$

where the Kronecker δ selects the pixel $\mathbf{x}_k$. The LEGM states that, assuming brightness constancy during a small $\Delta t = t_{N_e} - t_1$, the increment (6.7) is caused by brightness gradients ∇L moving with image velocity $\mathbf{v}$[44]:

$$\Delta L(\mathbf{x}) \approx -\nabla L(\mathbf{x}) \cdot \Delta\mathbf{x} = -\nabla L(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) \Delta t. \tag{6.9}$$

6.4.2 Optimization Objective

We cast the problem of estimating the displacement (6.6) as an optimization one, where we minimize the mismatch between the event data (in the form of (6.8)) and its prediction $\Delta\hat{L}$ via (6.9) exploiting the knowledge of the background pattern from a frame $\hat{L}$. This idea is

summarized in Fig. 6.5.

To allow for the fact that $\hat{L}$ may not be perfectly aligned with the corresponding events, we augment the model (6.9) with a translation warp $\hat{L}(\mathbf{W}(\mathbf{x};\mathbf{p}))$, where $\mathbf{W}(\mathbf{x};\mathbf{p}) = \mathbf{x} + \mathbf{p}$, and $\mathbf{p}$ denotes a small per-pixel translation.

Our composite objective (i.e., loss) function implies a joint optimization over the flow and alignment parameters:

$$E(\mathbf{v},\mathbf{p}) \doteq E_{\text{data}}(\mathbf{v},\mathbf{p};\mathcal{E}) + E_{\text{reg.}}(\mathbf{v},\mathbf{p};\mathcal{E}). \tag{6.10}$$

The data-fidelity term measures the goodness of fit between the event data $\mathcal{E}$ and its prediction with our model:

$$E_{\text{data}} \doteq \left\| \frac{\Delta\hat{L}}{\|\Delta\hat{L}\|_2}(\mathbf{x}) - \frac{\Delta L}{\|\Delta L\|_2}(\mathbf{x}) \right\|_\gamma, \tag{6.11}$$

where γ is the L^1 norm (robust norm). Since C in (6.8) is unknown, we compute the difference between normalized brightness increments (norms are over the pixel domain Ω).

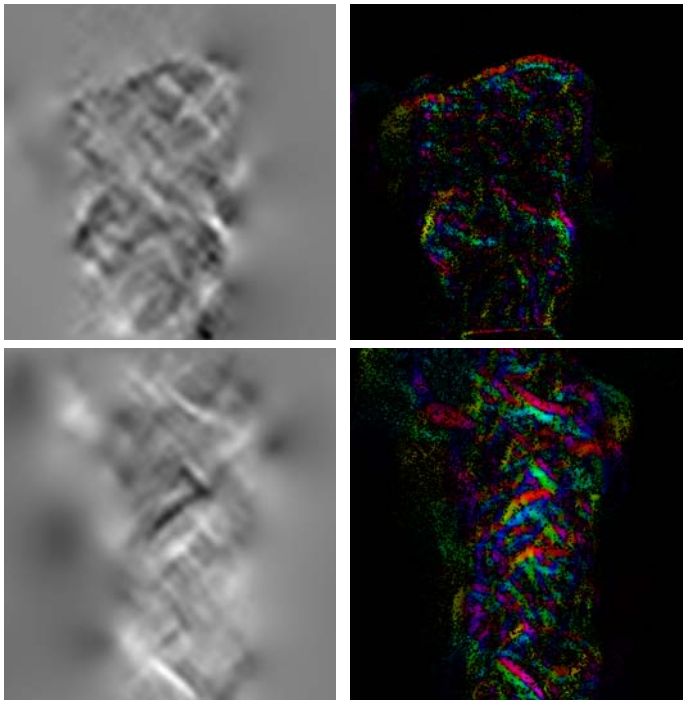
The regularizer penalizes the non-smoothness of the flow $\mathbf{v}$ and the magnitude of the per-pixel translation $\mathbf{p}$:

$$E_{\text{reg.}} \doteq \lambda_1 \|w(\mathbf{x})\, \nabla\mathbf{v}(\mathbf{q}(\mathbf{x}))\|_1 + \lambda_2 \|\mathbf{p}(\mathbf{x})\|_1. \tag{6.12}$$

The flow regularizer (first term in (6.12)) is explained in Sec. 6.4.4, after the flow parameterization is introduced. For the second term, the magnitude of $\mathbf{p}$ is given by its L^1 norm over the pixel domain. In the experiments, we set $\lambda_1 = 0.5$ and $\lambda_2 = 0.1$.

6.4.3 Physically-motivated Parameterization

Swapping the mixed derivatives (Schwarz's theorem) in (6.6), the flow $\mathbf{v} \sim \frac{\partial}{\partial\mathbf{x}}\frac{\partial\rho}{\partial t}$ is interpreted as the spatial gradient of $\frac{\partial\rho}{\partial t}$. Thus (6.6) admits two interpretations. (*i*) from left to right: once estimated, the flow may be Poisson-integrated [8] to obtain $\frac{\partial\rho}{\partial t}$, (as the best L^2 fit to the estimated flow[79,173]). (*ii*) from right to left: the flow may be obtained as the spatial (e.g., Sobel) gradient of a precedent scalar field $\frac{\partial\rho}{\partial t}$. In contrast to most optical flow estimation methods, which parametrize $\mathbf{v}(\mathbf{x})$ directly in terms of its x and y components, we go one step further and exploit the above second interpretation of (6.6) to parametrize the flow by means of $\mathbf{q} \equiv \frac{\partial\rho}{\partial t}$, which we call the Poisson parameters of the flow. This not only reduces the complexity of the problem (number of variables being optimized), thus conferring robustness but also provides a strong link with the physical meaning of the variables: according to (6.6), the resulting flow



(a) Poisson images **(b)** Optical flow

Figure 6.6: Poisson parameters and flow.

actually represents the schlieren objects. Figure 6.6 shows examples of the Poisson parameters $\mathbf{q}$.

Figure 6.5 summarizes the visual quantities involved in the calculation of (6.11). The candidate scalar parameter field $\mathbf{q}$ is converted (via the Sobel operator) into the vector flow field $\mathbf{v}$. The flow $\mathbf{v}$ and translation field $\mathbf{p}$ are used in the augmented model of (6.9) to generate a predicted (i.e., modeled) brightness increment image. On the other hand, events $\mathcal{E}$ are summed in (6.8) and Gaussian-smoothed to produce a measured brightness increment image. The difference between the measured and predicted brightness increments provides an error signal that is used to drive the iterative refinement of the unknown variables $\mathbf{p}$ and $\mathbf{q}$.

6.4.4 Flow Regularizer

We penalize the non-smoothness of the flow using a weighted Total Variation (TV) (see (6.12)). As illustrated in Fig. 6.4b, it is difficult to estimate accurate flow in regions with very few events, which correspond to constant (e.g., zero) flow, hence we impose this prior knowledge as a regularizer to encourage zero flow therein. Specifically, from the events $\mathcal{E}$ we

compute a Gaussian-smoothed histogram $h(\mathbf{x}; \mathcal{E}) = \sum_k \mathcal{N}(\mathbf{x}; \mathbf{x}_k, \sigma^2)$ (with $\sigma = 5$ px) and normalize it to the range $[0, 1]$. Then, we define weight function $w(\mathbf{x}) \doteq 1 - \alpha / h(\mathbf{x}; \mathcal{E})(\mathbf{x})$ (large in ill-posed regions with very few events), with $\alpha = 0.95$ in the experiments.

6.4.5 Optimization

Multi-scale. For improved convergence, a coarse-to-fine patch-based approach is used for $\mathbf{v}$, $\mathbf{p}$ and the loss function (6.10). The coarsest patch size is 64×64 px and we use four resolution levels in a pyramidal fashion, resulting in finest patches of 8×8 px. To reach pixel density from the finest patches, we use bilinear interpolation.

Implementation. As an optimizer, we use Adam[83] with 600 iterations. The learning rate is set to 0.05, with a decay of 0.1. The initialization of the first frame at the coarsest scale is: zero for $\mathbf{p}$ and $\mathbf{v}$ (when applicable) and random in $[-1, 1]$ for the Poisson parameters $\mathbf{q}$. We found the latter to be better than also setting $\mathbf{q}$ to zero. Then, the initialization of the next levels uses the optimization results from the previous scales (i.e., coarse-to-fine approach).

6.5 Physical Setup and Data

6.5.1 Recording Setup

Co-capture System. To achieve high-quality recordings of frames and events, we build our own acquisition system. Although some devices exist that record colocated events and frames (such as DAVIS[13,158]), their data quality (resolution, dynamic range, etc.) is limited and not suitable for BOS applications. Our custom-built co-capture system consists of a frame camera (Basler acA1300-200um, 1280×1024 px) and the latest generation event camera (Prophesee EVK3 Gen4, 1280×720 px[41]), sharing the same optical axis by using a beamsplitter (Plate Bs C-Mount VIS50R/50T). Both cameras are hardware-triggered for accurate synchronization and are calibrated to achieve accurate pixel alignment, following[107]. Figure 6.4 shows the camera system and an example of acquired data. Further details about the used recording system can be found in[66].

Optical Setup. The field of view (FOV) of our cameras is limited by the beamsplitter ($\approx 15°$), hence we set the distance between the cameras and the background to 3.3 m. We use randomly-generated background patterns that cover the whole FOV, where black dots (covering approximately 2 to 3 pixels in the image plane) are printed on white paper.

The data quality also depends on the distance between the camera and the schlieren object. The schlieren are more visible (larger pixel displacement $\Delta\mathbf{x}$) by keeping Z_A small (object closer to the camera). At the same time, the camera system has to be focused both on the background pattern and the schlieren object, thus Z_A cannot be too small. We experimentally found distance $Z_D = 1.6$ m to be a good compromise between both opposing effects. To

Table 6.2: Parameters of the recorded sequences.

Sequence	Convection	Luminance [lx]	Duration [s]	Event rate [Mev/s]
Hot plate 1	Natural	4000	19.4	11.3
Hot plate 2	Natural	225	19.8	5.1
Hair dryer (OFF) 1	Natural	4000	13.5	5.1
Hair dryer (OFF) 2	Natural	4000	19.7	5.3
Hair dryer (OFF) 3	Natural	225	14.7	2.8
Crushed ice	Natural	4000	17.4	5.0
Hair dryer (ON)	Forced	4000	13.4	15.0
Breathing 1	Forced	4000	12.8	4.0
Breathing 2	Forced	4000	13.0	3.7

control the scene brightness and achieve uniform illumination in the background, we use LED panels (four Eurolite LED PLL-360). This illumination allows us to lower the aperture to an f-number of 10, leading to a higher depth of field. Note that our beamsplitter setup leads to a 50% split of the light reaching each camera of the acquisition system.

6.5.2 Data Acquired

We record multiple sequences with natural and non-natural (forced) air convection, which are summarized in Tab. 6.2. For natural convection, we use heat sources, such as a hot plate, a hair dryer (switched off), and ice. To demonstrate the HDR capabilities of event cameras, we record the data in (*i*) bright conditions ($\approx$ 4000 lx) and (*ii*) low-light conditions ($\approx$ 225 lx). The low-light condition is set to be darker than normal office lighting, which is a more natural condition for real-world applications.

Each sequence is approximately 10 to 20 seconds long and consists of events, frames, and a calibration parameter file. The recording starts with the scene in the absence of the schlieren object, which is useful for frame-based BOS methods (reference frame). All sequences are recorded at normal room temperature ($\approx$ 24°C). For the forced convection sequence of the running hair dryer, we set the event camera's refractory period to its minimum possible value to capture the fast dynamics of the airflow. In total, we record nine sequences, each of which has up to 200M events.

Frames of sample sequences are shown in Fig. 6.7. Each frame is mapped from its original resolution (1280$\times$1024 px) to the event-camera resolution (1280$\times$720 px) (see Sec. 6.5.1).

Since we cannot obtain real ground truth (GT), we use frame-based estimated flow as GT flow (Fig. 6.7). The calculation of the flow is based on the classical Farnebäck algorithm[40]

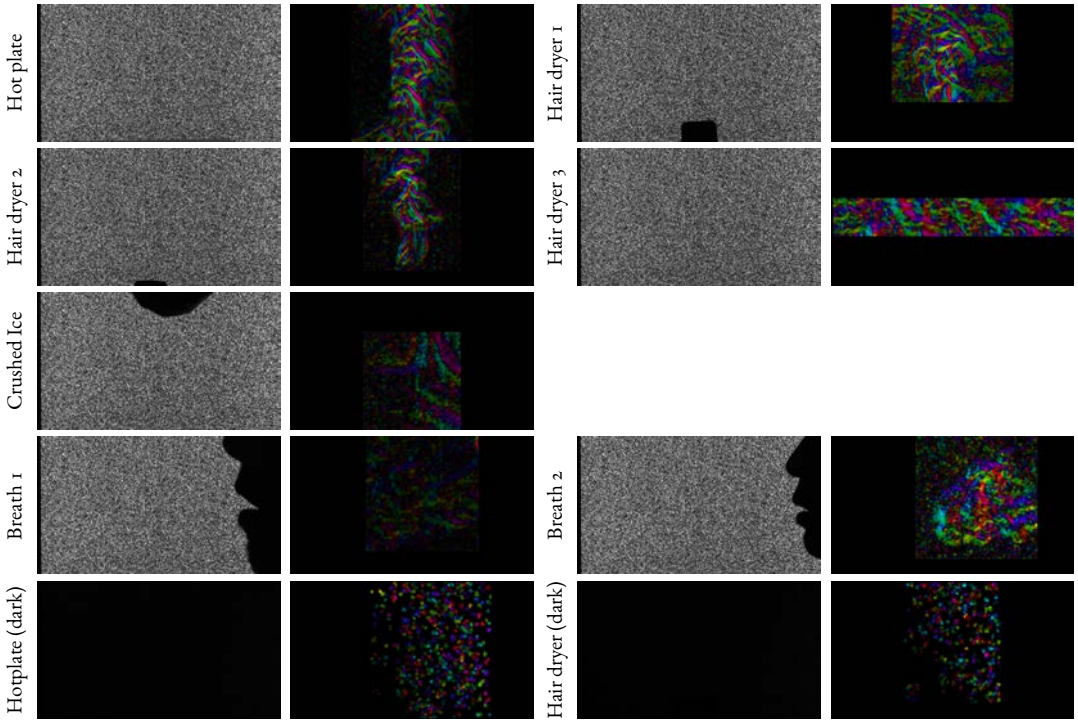
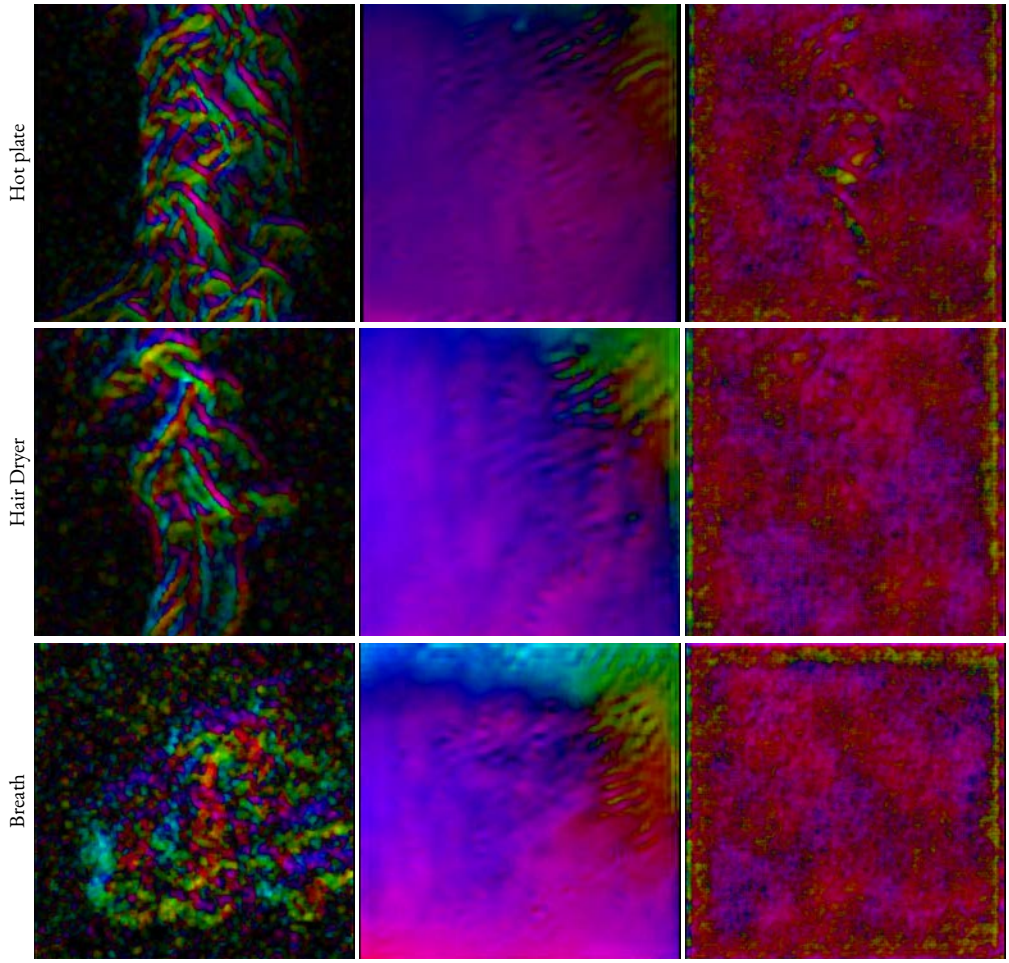


Figure 6.7: *Sample frames from each sequence and the frame-based flow.* Frames mapped into the event-camera image plane are shown on the left. The estimated optical flow (inside the ROI) is shown on the right. For the low-light sequences, the frame-based method fails to estimate reasonable flow. Nevertheless, we show them for completeness.

with four pyramidal scales at the frame rate (120 fps). We test different parameters and find no significant difference on the quality of the results. Before settling for Farnebäck's algorithm, we tested recent DNN-based state-of-the-art methods, such as[74,75], and found that they do not produce reasonable flow. Figure 6.8 shows the comparison of several frame-based optical flow estimation methods: two state-of-the-art optical flow and video-frame interpolation works[74,75] and Farnebäck's method. Due to the large gap between the training datasets of[74,75] and our dataset, these recent DNN-based methods fail to estimate reasonable flow. Farnebäck's algorithm works robustly and better, because (*i*) the background pattern is parallel to the image plane, (*ii*) the scene has no occlusions, (*iii*) the background pattern has clear and random edges that are useful to calculate the deformation between two frames. Since we cannot determine the real GT, we do not explore a further analysis of frame-based estimation methods, which we leave for future research, such as simulation. That is, to establish the first event-based BOS problem settings we leverage the knowledge of established frame-based BOS techniques. Note that the quantitative evaluation is only based on the well-illuminated sequences since the frame-based flow degrades in dark scenes (see Sec. 6.6.6). We publish the

dataset and the code to compute the GT.



(a) Farnebäck's[40] **(b)** FlowFormer[74] **(c)** RIFE[75]

Figure 6.8: Different frame-based optical flow methods.

6.6 Experimental Evaluation

This section reports the performance of the proposed estimation method and its properties. First, we explain the baseline methods and evaluation metrics (Sec. 6.6.1). Second, we benchmark the accuracy of all methods considered (Sec. 6.6.2). Third, we show the capabilities of our method in low-light conditions (Sec. 6.6.3) and how it achieves high temporal resolution (1200 Hz "slow motion") in Sec. 6.6.4, including a velocimetry application (Sec. 6.6.5).

Table 6.3: Details of the benchmark. "ROI position" contains the coordinates of the top-left corner.

Sequence	ROI size [px]	ROI position [px]	Duration [s]	Total events
Hot plate 1	640×720	[320, 0]	10 to 14	51 900 802
Hot plate 2 (dark)	640×720	[420, 0]	12 to 14	12 912 262
Hair dryer (OFF) 1	640×640	[320, 0]	4 to 7	13 498 252
Hair dryer (OFF) 2	512×640	[384, 0]	6 to 7	4 089 883
Hair dryer (dark)	512×640	[384, 0]	5 to 7	3 460 579
Crushed ice	512×512	[384, 208]	8 to 11	5 856 190
Hair dryer (ON)	1280×200	[0, 260]	3.3 to 4.3	17 860 129
Breathing 1	590×600	[400, 0]	4.36 to 5.5	2 783 122
Breathing 2	640×640	[447, 0]	2.5 to 3.5	1 811 889
Total	–	–	18.14	114 173 108

Finally, we analyze the proposed method further, especially regarding the dependency on frames (Secs. 6.6.6 and 6.6.7) and its sensitivity to hyper-parameters (Sec. 6.6.8).

6.6.1 Evaluation Metrics and Baseline Methods

Evaluation Metrics. We evaluate the proposed method in terms of optical flow $\mathbf{v}$ accuracy. Two variants of the method are assessed: (*i*) using $\mathbf{q}$ as parameterization, from which we obtain $\mathbf{v}$ afterwards via (6.6), and (*ii*) using $\mathbf{v}$ directly.

The optical flow evaluation metrics are the average endpoint error (AEE), the percentage of pixels with AEE $>$ 1 px (denoted by "% Out"), and the angular error (AE). We select the time interval (from 1 to 4 s) and region of interest (ROI) to remove objects, such as a hair dryer and a face from the scene. All metrics are computed over pixels with at least one event inside the ROI.

Table 6.3 reports the detailed duration, ROI, and the total number of events used for the benchmark. The duration is selected such that the quality of schlieren is the best and most stable. For the "Hair dryer (ON)" sequence, we limit the height of the ROI due to the extremely large number of events observed: otherwise, we set the ROI to have approximately 720×512 px.

Baselines. As baseline flow estimators we use the two self-implemented methods from events because, to the best of our knowledge, there are no methods that estimate schlieren flow from event camera data.

- The Multi-reference Contrast Maximization (MCM)[146] (Chapter 4) is a state-of-the-art optical flow estimation algorithm from events alone. It is a model-based method,

Table 6.4: Results of optical flow estimation.

	Hair dryer (OFF) 1			Hair dryer (OFF) 2			Hot plate 1			Hair dryer (ON)		
	AEE ↓	%Out ↓	AE ↓	AEE ↓	%Out ↓	AE ↓	AEE ↓	%Out ↓	AE ↓	AEE ↓	%Out ↓	AE ↓
MCM [146]	1.425	35.639	0.621	0.421	10.886	0.476	**0.400**	21.789	0.426	0.287	5.933	0.712
E2VID [131]	1.055	39.068	0.677	1.091	37.734	0.670	1.092	32.121	0.611	0.811	25.997	0.587
Ours (Flow)	0.675	22.104	0.404	0.688	24.930	0.448	0.810	30.289	0.544	0.310	6.756	0.258
Ours (Poisson)	**0.383**	**9.319**	**0.299**	**0.395**	**10.174**	**0.337**	0.487	**12.215**	**0.421**	**0.215**	**0.924**	**0.202**

	Crushed ice			Breathing 1			Breathing 2		
MCM [146]	1.090	96.964	0.823	1.769	49.552	0.853	2.056	78.690	0.973
E2VID [131]	1.249	55.030	0.791	1.014	42.072	0.692	1.056	43.348	0.699
Ours (Flow)	0.587	21.815	0.452	0.665	11.872	0.341	0.557	17.716	0.438
Ours (Poisson)	**0.326**	**5.177**	**0.301**	**0.345**	**6.322**	**0.203**	**0.476**	**8.028**	**0.410**

hence there is no mismatch in the training dataset (due to our specific background pattern). We use the events between two consecutive frames (i.e., in a time span of 8.3 ms).

- Flow estimation from reconstructed intensity images: we use E2VID [131] (a learning-based approach) to compute grayscale images from events and then apply the same (frame-based) optical estimator as the one for the GT. Images are reconstructed at 120 fps, i.e., the same frequency as the frames.

To the best of our knowledge, we found no methods with publicly-available implementation combining events and frames to estimate the optical flow, we, therefore, believe this is a best-effort comparison. Also, notice that we do not train a Deep Neural Network (DNN) model with the supervisory GT flow, as the purpose of the chapter is not a purely data-driven approach, but to develop an interpretable model-based method, by deriving a connection between the physical parameters and the data.

6.6.2 Optical Flow Evaluation

Flow accuracy is reported in Tab. 6.4. We evaluate on illuminated sequences for valid GT flows from frames (please see Sec. 6.6.3 for the dark sequences). Consistently for almost all sequences, the proposed method ("Ours (Poisson)") provides the best accuracy compared with the baseline methods. Due to the nature of schlieren, the GT flow magnitude has normally subpixel values. Hence, we find that the angular error (AE) is a more reliable metric for the purpose of this benchmark. The largest magnitude of the displacement ($\approx$ 3 px) is observed in the hotplate sequences. Still, it is remarkable that the proposed method achieves AEE $<$ 1 pixel. We acknowledge that the proposed method utilizes both event and frame data, while the baselines use only event data as input. This is further discussed in Sec. 6.6.6.

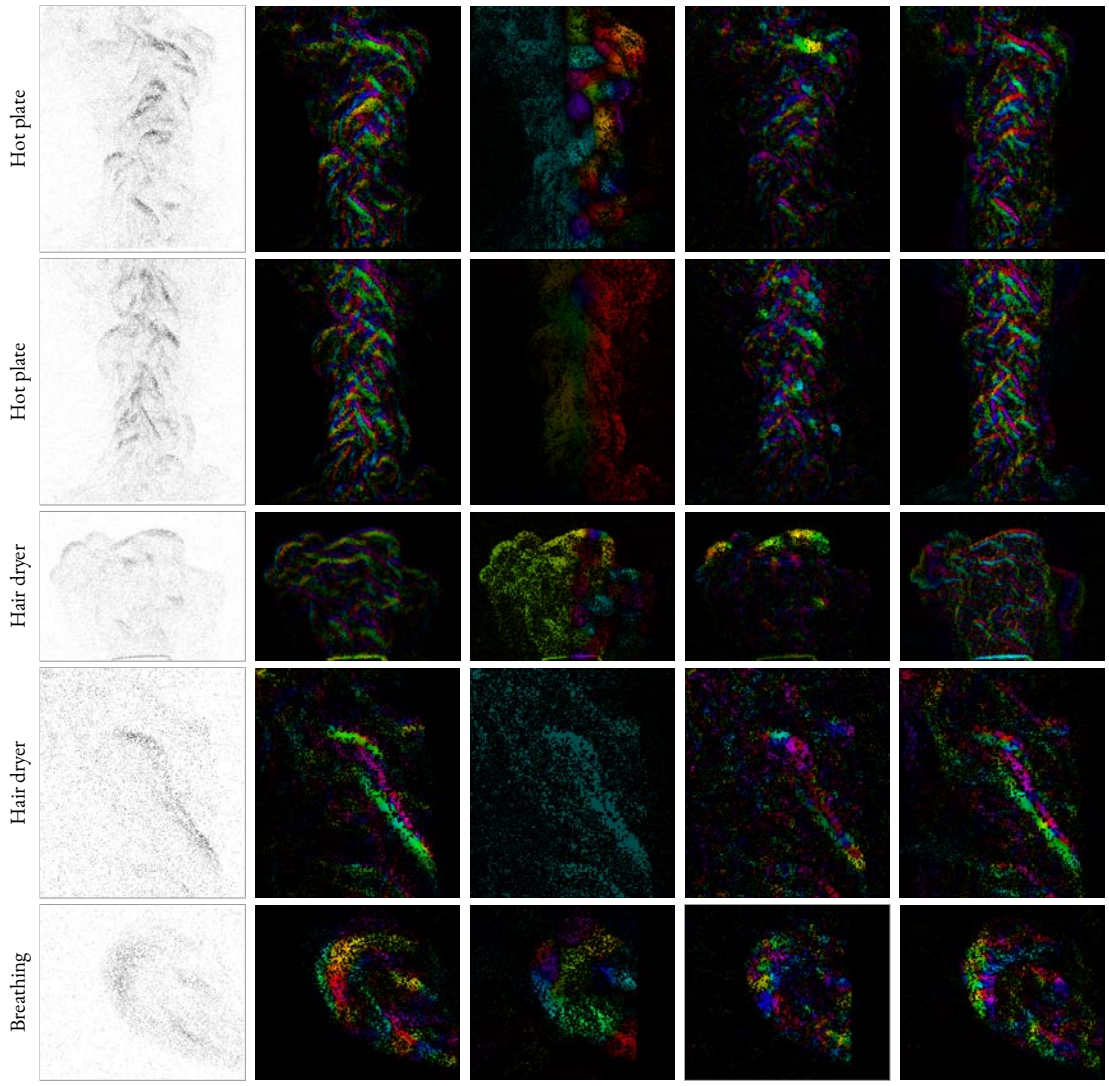


Figure 6.9: Qualitative comparison between different flow estimation methods.

Also, it is noticeable that the Poisson-parameterized estimation ("Ours (Poisson)") results in better accuracy than the flow-parameterized estimation ("Ours (Flow)"). This clearly states the effectiveness of our physically-motivated parameterization. It provides not only a smaller number of parameters, as discussed in Sec. 6.4, but also contributes with better accuracy.

Additionally, we observe that forced convection usually has a smaller displacement magnitude than natural convection. This is because the optical flow $\mathbf{v}$, which we evaluate on, is

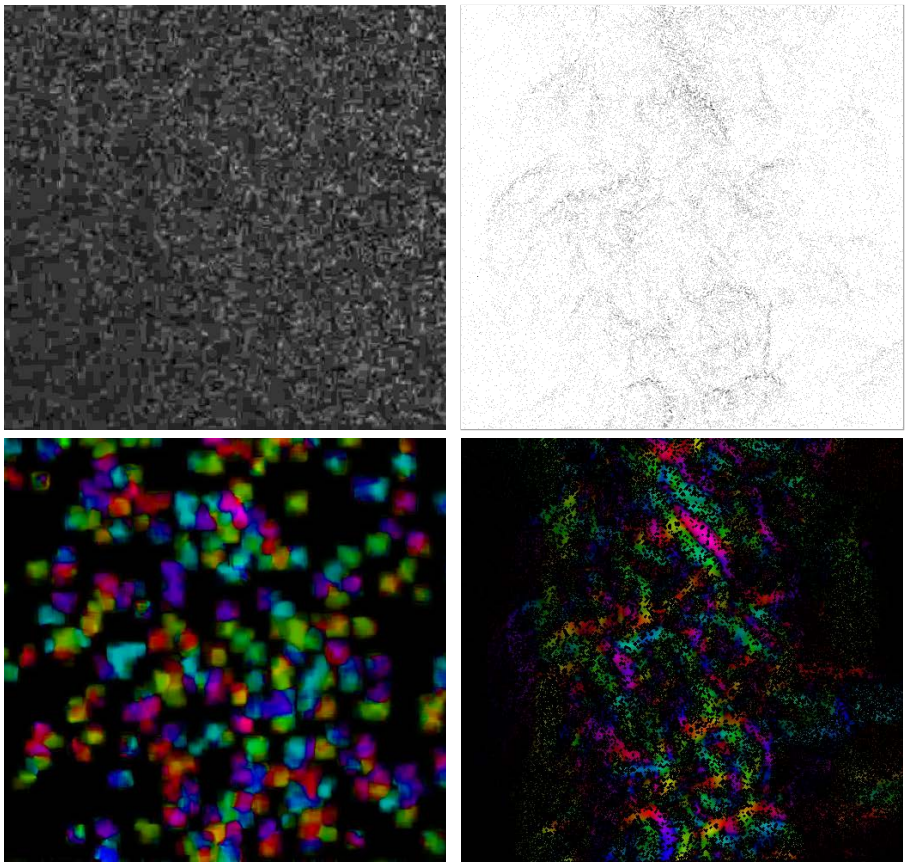
the temporal derivative of the density gradient. In the forced convection case (e.g., hair dryer (ON)), the spatio-temporal changes of the air density at a pixel might be smaller than in the natural, heat-induced schlieren, since the advection of the flow is dominant, which can be seen as nearly constant.

Figure 6.9 shows qualitative results. Although the GT flow is based on a classical, general-purpose estimation method, it provides remarkably reasonable flow. The baseline methods (MCM and E2VID) fail to estimate reasonable flow from events. Especially, we find the alignment-based method[146] fails to estimate schlieren flow. This is because most events are generated at the edges of the background pattern, resulting in an uneven spatial distribution despite air is actually moving, and consequently, triggering more flickering events. The E2VID-based method surprisingly reconstructs edge structures of the background pattern (see also Sec. 6.6.7) in spite of this specific (flickering) event input and estimates comparable flow. However, it fails to recover the fine structure of the flow. Finally, the flow estimated by our method resembles the GT flow the most, and it even seems to capture more fine-scale (high-frequency) structures.

6.6.3 HDR Experiment

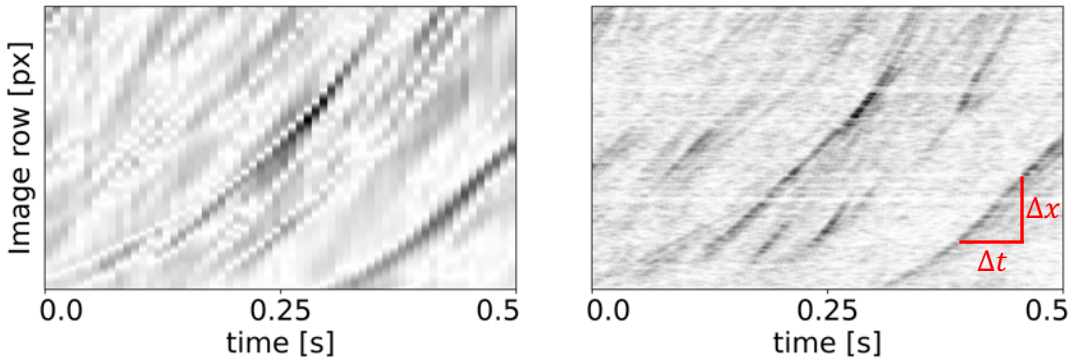
So far we have established that the proposed method is able to recover the fine flow structure of the schlieren object. However, schlieren based on events has another interesting aspect: as shown on the left column of Fig. 6.9, the existence of schlieren is already visible in the event data histogram. By contrast, the schlieren structure is not visible to the naked eye on the raw frame data but only as the result of optical flow processing. The fact, that schlieren is observable in a more direct way using events, allows us to leverage the advantages of the event camera itself, such as HDR and high temporal resolution.

Figure 6.10 shows qualitative results of the frame-based and event-based schlieren imaging under poor illumination. The frame-based schlieren method fails to estimate realistic flow under such conditions, as it needs intense lighting sources, especially if high-speed cameras are used. Due to the insufficient brightness, the quality of the frames collapses even after normalization (i.e., using the entire grayscale range). On the other hand, the event data capture the schlieren structure (Fig. 6.10, top right). Furthermore, the proposed algorithm combining events and frames is surprisingly robust against such low-quality image inputs. Using natural light (225 lx) the result (Fig. 6.10, bottom right) shows the potential of event cameras to push the limits for future BOS applications. We further discuss the effect of the amount of illumination in Sec. 6.6.6.



(a) Using normalized frames **(b)** Ours

Figure 6.10: *Schlieren imaging under low illumination (HDR).* (a) Frame-based methods suffer from the limited dynamic range of the frames, resulting in unrealistic flows with artifacts despite using all grayscale range available for the frames (normalization). (b) The proposed method produces a realistic flow, similar to the event data, which is visible due to the HDR nature of events.



(a) Frame-based kymogram

(b) Event-based kymogram

Figure 6.11: *Kymograms (space-time plots)* for a 0.5 second excerpt of a hotplate sequence. (a) The frame-based schlieren imaging is limited to the temporal resolution of the camera (120 Hz). (b) The event-based schlieren can recover higher temporal resolution (e.g., 1200 Hz) thanks to its data property.

6.6.4 Super-Slow Motion

Event-based BOS also enables us to see the schlieren at markedly higher temporal resolution (i.e., slow motion) than conventional frames. To this end, we conduct a streak-schlieren analysis [144]. The streak analysis focuses on a single column of the schlieren image to see how it evolves in time, by showing an $x - t$ diagram (kymogram) of the air convection. The frame-based schlieren method uses for example Poisson images as schlieren images. For event-based methods, schlieren images can be either Poisson images or simply event histograms. Figure 6.11 shows a comparison of kymograms obtained from frames at 120 Hz (the frame rate) and obtained from events (10× higher rate, i.e., at 1200 Hz). Event-based BOS can provide high temporal resolution kymograms due to the asynchronous nature of event data. Compared with the frame-based analysis (Fig. 6.11a), the event-based one (Fig. 6.11b) shows thinner lines of schlieren in space-time. The slow motion schlieren visualization is best viewed in the supplementary video.

6.6.5 Velocimetry

One can perform velocimetry by fitting curves to the kymograms [144]. Let us analyze the speed of propagation of schlieren ($\partial\rho/\partial t$ in the case of Poisson image) along one direction (e.g., vertical). Figure 6.11b shows an example on the hot plate sequence. By fitting a curve (line), the flow propagates 166 pixels during approximately 68.8 milliseconds. The geometry of the BOS setup (focal length $f = 25$mm, distance to object $Z_A = 1.7$m, pixel size 4.86μm) leads to an approximate velocity of 0.805 m/s.

6.6.6 Dependency on Frames

The proposed method uses events and frames. Naturally, the question arises to which extent the algorithm relies on which signal. To this end, we present the ablation study with different brightness levels (see also Sec. 6.6.3). Figure 6.12 shows the qualitative results for both: the frame-based method and our method (frame plus events), for different illumination levels (measured with a Voltcraft MS-1300 light meter). As clearly shown, the frame-based flow (column (b)) starts to deteriorate when the illumination is 1000 lx or smaller. For better performance, we even normalize the range of the frames used (the exposure time is fixed to maintain the frame rate of 120 fps). However, this does not provide significantly better results that can compete with those of our method. By contrast, the following two points are remarkable about our method: (*i*) schlieren is still visible at 110 lx in the event histograms, indicating the HDR capabilities of the noisy input data (column (c)), and (*ii*) the estimated flow (column (d)) still looks reasonable when the illumination is as low as 225 lx, despite our method using the naturally darker frame as an input (column (a)). Note that our method does not work when the frame is completely black (less than 50 lx). All the above indicates that the proposed method requires frames, but it can overcome the limited dynamic range of the frames due to the HDR advantages of event cameras.

6.6.7 Towards a Frame-Free Method

The proposed method utilizes the information from events and a frame, however, the quality of the frame data does not need to be the best, as shown in the previous section. Hence, an interesting challenge is to replace frame data with intensity reconstruction from events, such that the proposed method could be extended to be *frame-free*. To this end, Figure 6.13 shows the comparison of the different input frames. Instead of using an acquired frame as an input to the proposed method, we reconstruct intensity images using E2VID[131] and feed them as input. Despite the large visual difference between the two different inputs, the output flow and Poisson images seem to have similar structures. Although we do not further investigate the quality of the intensity reconstruction, the results show future possible extensions toward frame-free event-based BOS methods.

6.6.8 Effect of the Regularizers and the Translation Field

Ablation. To assess the importance of the regularization and the translation field parameters $\mathbf{p}$, we conduct an ablation study. The top half of Tab. 6.5 reports the optical flow accuracy of the proposed method, the one without regularization, and the one without the translation. There is a significant improvement due to the regularizers: without regularizers, the estimated

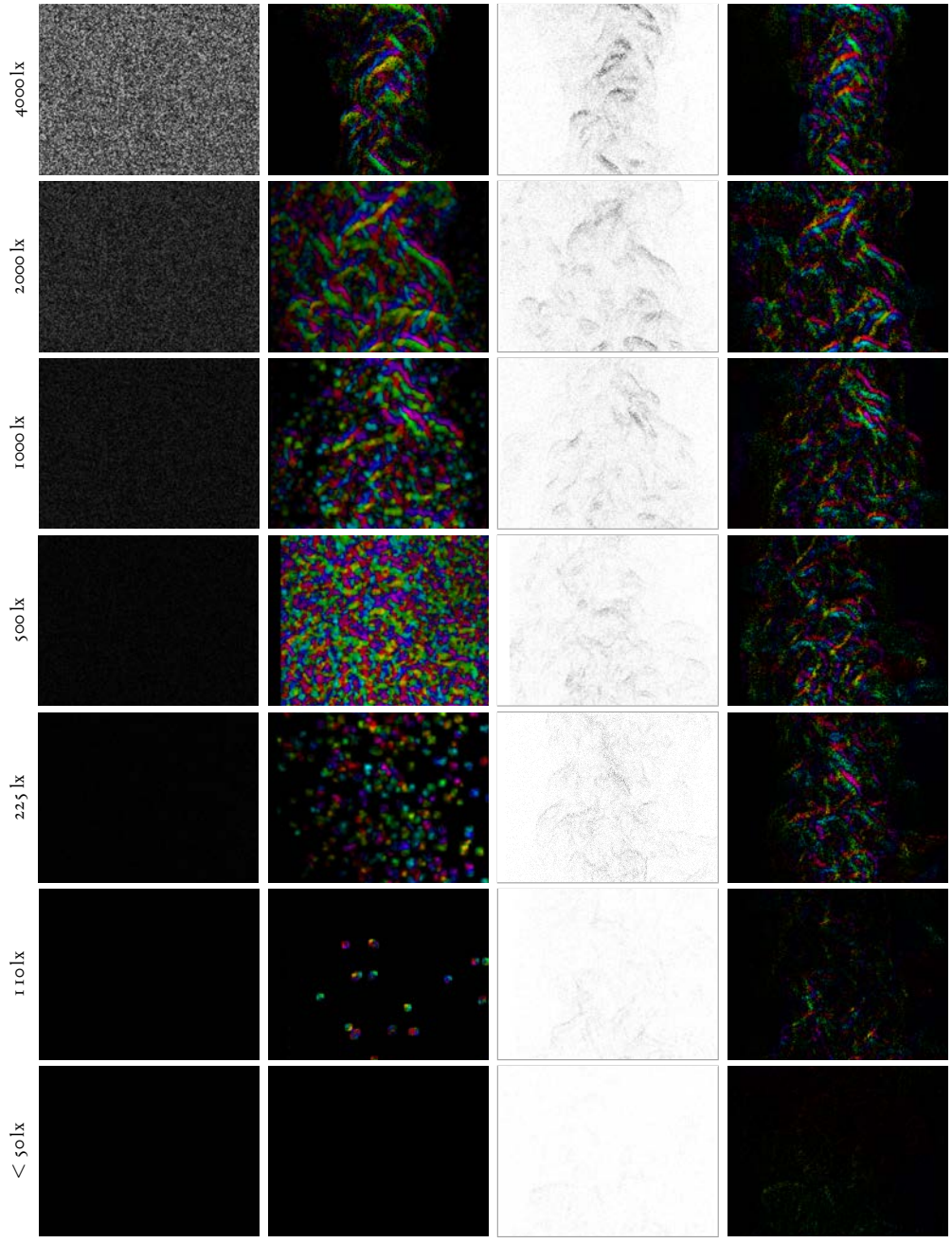
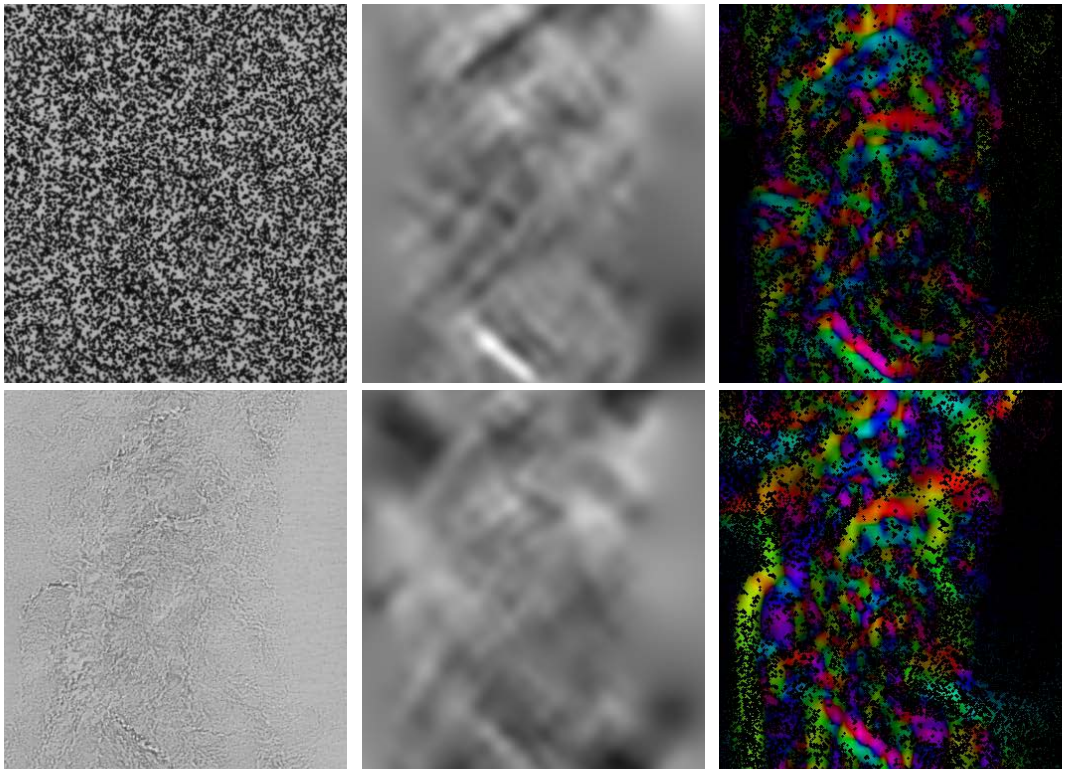


Figure 6.12: *Ablation study for different lighting conditions.* Flow (b) uses normalized frames as input, while our flow (d) uses events (c) and original frames (a).



(a) Input **(b)** Output Poisson **(c)** Output Flow

Figure 6.13: *Towards a frame-free method.* The top row shows the originally proposed method with the frame-based camera input. The bottom row shows an E2VID-reconstructed image as the alternative input. In spite of the large quality difference between the two inputs (a), the output Poisson and flow images have some visual similarities (b,c).

q and **p** become not smooth anymore, which leads to irregular flow estimation. The effect of **p** is relatively minor but still noticeable.

Sensitivity Analysis. We test different weights for each regularizer λ_1, λ_2 in (6.12). The weights are set as follows: we fix one parameter ($\lambda_1 = 0.5$), and vary λ_2 between 0.01 and 1.0; then we fix the other parameter ($\lambda_2 = 0.1$) and vary λ_1 between 0.05 and 1.0. The flow accuracy is reported in the bottom half of Tab. 6.5. We observe on-par accuracy when $\lambda_1 = 1.0$ with respect to the base condition (the top row).

6.7 Limitations

The proposed BOS technique using events shows advantages over frame-based BOS in terms of HDR capabilities and temporal resolution, lowering the demand for bright illumination

Table 6.5: Results of the ablation study and the sensitivity analysis.

	Hot plate			Crushed Ice			Dryer		
	AEE ↓	%Out ↓	AE ↓	AEE ↓	%Out ↓	AE ↓	AEE ↓	%Out ↓	AE ↓
Ours ($\lambda_1 = 0.5, \lambda_2 = 0.1$)	0.487	12.215	0.421	0.326	5.177	0.301	0.395	10.174	0.337
w/o regularizers (i.e., $\lambda_1 = \lambda_2 = 0$)	3.371	82.039	1.111	2.499	76.325	1.017	1.233	48.626	0.756
w/o translation model (i.e., $\mathbf{p} = 0$)	0.591	18.609	0.488	0.368	7.791	0.313	0.394	10.896	0.324
$\lambda_1 = 0.05, \lambda_2 = 0.1$	0.586	14.468	0.494	0.518	11.295	0.440	0.449	11.104	0.387
$\lambda_1 = 1.0, \lambda_2 = 0.1$	0.482	10.462	0.416	0.390	3.849	0.349	0.378	7.274	0.330
$\lambda_1 = 0.5, \lambda_2 = 0.01$	0.509	11.001	0.440	0.437	5.482	0.386	0.398	7.129	0.344
$\lambda_1 = 0.5, \lambda_2 = 1.0$	0.517	11.598	0.443	0.429	5.609	0.379	0.409	8.112	0.350

and high-speed cameras. However, in other aspects, it inherits the limitations of frame-based BOS. Optically, the estimated brightness gradient is a mean value integrated along the optical axis, and the technique inherently has a trade-off between the observed displacement and the obtained sharpness of the gradient under investigation.

Additionally BOS is sensitive to vibrations, due to the underlying assumption that the small perceived changes are only caused by refractive index variations. Specific to event cameras is that the signal is noisy, and careful tuning of the camera's biases is necessary. The proposed method furthermore relies on a combination of events and frames, thus an accurate spatio-temporal alignment of both data sources is required. The flow estimation method does not run in real time. However, raw events visualized as histograms can be computed online and resemble schlieren images. While the proposed multi-scale approach improves the convergence of the optimization, it limits the spatial resolution of the flow, which is a similar limitation as in frame-based BOS.

6.8 Conclusion

In this chapter, we have presented the first event-based BOS imaging and an algorithm to estimate the temporal derivative of the air density gradient. The approach has been mathematically rigorously obtained and has a physically-motivated parametrization. Using the frame-based method as GT the experiments evidenced that our approach outperforms all other tested methods. We furthermore illustrated how the advantages of event cameras could be leveraged for BOS applications, lowering the requirements for high illumination and visualizing the turbulent eddies at a significantly higher temporal resolution. We release the code and dataset to the public and hope that this research opens up new possibilities for the computer vision community.

7 Conclusion

7.1 Summary

This thesis focused on various motion estimation problems within short time intervals using a single event camera. The research questions tackled were:

- How can we extend CMax for broader types of motion hypotheses by improving the objective function?
- How can we take the space-time nature of events into account to rethink event-based optical flow?
- What is a more biologically-plausible solution for event-based optical flow?
- How can we utilize event-based motion estimation in imaging science to leverage the event camera advantages?

In Chapter 3, we focused on the low-DOF (ego-motion) estimation problems, up to 8 DOFs, and improved the well-posedness (the objective function landscapes) in the Contrast Maximization framework. Here, the static scene was an approximation and hence its limitation. In Chapter 4, we focused on the high-DOF (optical flow) estimation problem. We presented a principled estimation method that effectively mitigates event collapse, handles occlusions better, and is transferable to unsupervised-learning settings. As opposed to the

previous two chapters, Chapter 5 proposed an event-by-event (incremental) method to estimate optical flow. The proposed triplet matching achieved high throughput (runtime per event), stemming from neuroscience. As an application of the event-based optical flow estimation, Chapter 6 demonstrated the capability of sensing air convection. In contrast to the previous chapters, here the task consisted of estimating the motion of air density using schlieren techniques by combining the complementary information of events and a frame. The extended linearized event generation model estimated the spatio-temporal derivatives of air density via optical flow computation.

7.2 Discussions and Future work

While the thesis has advanced knowledge in various topics, there are still pending problems and new research questions to be answered.

Event collapse. Event collapse has been tackled by adding regularizers in the low-DOF problems (Chapter 3) and by changing the data-fidelity part of the objective function in the high-DOF problem (Chapter 4). Although we showed the proposed methods are effective in the problem settings considered, event collapse could still appear in other scenarios. For example, the regularizers from Chapter 3 have not been investigated for the static scene assumption in its most complex form ("ideal" solution of the problem with 6-DOF ego-motion and N_p-DOF scene depth parameters). Also, the weight of the regularizer depends on the scene, since the data-fidelity term of the optimization (e.g., contrast functions) depends on the scene. To make the proposed regularizer even more effective, it would be desirable to make the regularizer weight independent of the scene.

The multi-reference focus loss proposed in Chapter 4 is not a "silver bullet" for event collapse. This is because the resulting landscape of the multi-reference objective function could be dominated by one (steep) optimum from one reference time. Nevertheless, it is important to further investigate the multi-reference idea, which essentially utilizes the property that the event stream is asynchronous with a high temporal resolution. Also, combining the proposed methods in Chapters 3 and 4 for optical flow estimation would be worth investigating, by calculating numerical approximations of the regularizers.

Optical flow in event-based vision. The proposed space-time (time-aware) flow in Chapter 4 extends the conventional frame-based optical flow that is a function of space. Recently, there have also been some proposals that aim to leverage the space-time nature of events for per-pixel motion estimation, such as[57,119]. The main difference between the space-time flow in Chapter 4 and proposals[57,119] is the motion hypothesis and its underlying assumptions: the space-time flow in Chapter 4 assumes that the flow is constant along its streamlines within

short time intervals, which results in linear trajectories of the warp ((4.8) and Fig. 4.3). The DOFs of the motion are still $2N_p$, and the efficacy of the parameterization for occlusions is shown in Sec. 4.3.5.

On the other hand, recent works [57,119] propose non-linear trajectories (e.g., Bézier curves) of the "optical flow", which results in an increased number of DOFs (larger than $2N_p$). The complexity of the motion estimation problem (the DOFs of the motion hypothesis) has a trade-off with the proneness of overfitting. Although it can be shown that the point trajectories in some low-DOF motion models are curves and not linear (e.g., 3-DOF rotational motion, see Fig. 3.3 and Sec. A.3), the efficacy of non-linear trajectories is yet to be investigated. We suspect that the choice of assuming non-linear trajectories in [57,119] stems from the necessity of reporting good figures on the DSEC benchmark [56]. However, note that while it is called the "optical flow" benchmark, the ground truth is provided over time intervals of 100 ms at moderate vehicle speeds, which results in non-linear trajectories. This conflicts with the classic definition of optical flow, which is the instantaneous velocity of the motion. The velocity only defines a straight line tangent to the curve, as opposed to a more complex non-linear trajectory. Therefore, one should reconsider the terminology of the motion-estimation task, such as "instantaneous" (optical flow, or short-baseline) vs. "non-instantaneous" (i.e., "large baseline" in frame-based vision) or curved trajectory estimation. This difference would also affect the definitions of the ground truth and evaluation metrics.

Accuracy and runtime. As shown in Chapters 4 and 5, there is a clear trade-off between accuracy and runtime in optical flow estimation. This partly comes from the design of the algorithm: handling events by batch or event-by-event (incrementally). Typically accumulating events over a certain time interval makes estimation more robust to noise and hence achieves better accuracy, while event-by-event methods provide fast but less accurate estimations. In the end, one needs to choose the trade-off (i.e., accumulation time for the method) depending on the optical flow application.

Also, a known limitation of the Contrast Maximization framework is that it requires some pixel displacement (typically 3 to 5 pixels) of the edges causing the events to have a good landscape of the objective function (optima at the desired motion parameters). Hence it is not robustly working in some cases, such as (*i*) a scene that has various displacements (e.g., various depths) and (*ii*) a scene where a robot is not significantly moving: one can not robustly guarantee sufficient displacement in the scene by choosing a constant size of the event batch. The triplet matching method (Chapter 5), on the other hand, can be interpreted as a method that asynchronously forces each edge to have a displacement of 3 pixels. In other words, the triplet matching method tries to require the shortest pixel displacement and to accumulate the displacements locally at every event, as opposed to the accumulation in the CMax-based methods whose window is defined globally. From the application point of view,

it will be important to tackle the challenge of these accumulation problems to try to improve the accuracy-runtime trade-off.

Flickering event. One of the challenges in event-based background-oriented schlieren (Chapter 6) was the flickering-form events from its specific recording settings. The previous optical flow estimation methods (Chapters 4 and 5) do not handle the flickering events, hence they fail to estimate realistic flow for such input data. In addition to the special recording setup, indeed, flickering events may happen in non-constant illumination scenarios, where the method in Chapter 4 deteriorates to some extent (e.g., night sequences in the DSEC benchmark[55]). It will be necessary to handle flickering events to produce more robust motion estimation from events.

Schlieren imaging. Chapter 6 demonstrated the great potential of event cameras in imaging sciences, especially towards high-speed and high-dynamic range recordings. Although we believe that schlieren imaging is an interesting application and will be another significant research field for event cameras, the lack of ground truth for BOS settings is an important challenge. To the best of the author's knowledge, there is no high-quality simulator that satisfies the complex modeling of the fluid dynamics and the optical mechanism of event cameras. The development of a simulator would enable further investigation and new methods, such as learning-based event-based BOS.

This thesis expands the understanding of event data and deepens event-based motion estimation by tackling its new data modality and challenges. We hope this work fosters future research and applications on event-based motion estimation.

A Warp Models

A.1 Preliminaries

In homogeneous coordinates, a homographic warp $\mathbf{W}$ is given by[46]

$$\mathbf{x}_k^{b\prime} \sim \mathtt{H}^{-1}(t_k;\theta)\,\mathbf{x}_k^b, \tag{A.1}$$

and the point trajectories of the warp are represented by

$$\mathbf{x}^b(t) \sim \mathtt{H}(t;\theta)\,\mathbf{x}^b(0). \tag{A.2}$$

Divergence. The flow is in Euclidean coordinates:

$$\mathbf{f} = \frac{\partial \mathbf{x}'}{\partial t}. \tag{A.3}$$

The flow divergence is given by $\nabla \cdot \mathbf{f}$.

Deformation. Using Result 1 in Appendix A of[128], the determinant of the Jacobian J of the transformation (from t_1 to t_k) in Euclidean coordinates is

$$\det\big(\mathtt{J}\big) = \frac{\det(\mathtt{H})}{(\mathbf{e}_3^\top \mathtt{H}\,\mathbf{x}^b(t))^3}, \tag{A.4}$$

where $\mathbf{e}_3 = (0,0,1)^\top$, $\mathbf{x}^b(t) = (x(t), y(t), 1)^\top$, and the Jacobian $\mathtt{J} = \frac{\partial \mathbf{x}'}{\partial \mathbf{x}}$.

Rate of change of area deformation. The differential transformation from t to $t + \Delta t$ is given also by a homography $\mathtt{H}_{t,t+\Delta t}$:

$$\mathbf{x}^b(t+\Delta t) \sim \underbrace{\mathtt{H}(t+\Delta t;\theta)\,\mathtt{H}^{-1}(t;\theta)}_{\mathtt{H}_{t,t+\Delta t}}\,\mathbf{x}^b(t). \tag{A.5}$$

Using the same notation as (A.4), the determinant of the Jacobian $\mathtt{J}$ of the transformation (from t_1 to $t + \Delta t$) in Euclidean coordinates is

$$\det\left(\mathtt{J}_{t,t+\Delta t}\right) = \frac{\det(\mathtt{H}_{t,t+\Delta t})}{(\mathbf{e}_3^\top \mathtt{H}_{t,t+\Delta t}\,\mathbf{x}^b(t))^3}. \tag{A.6}$$

The 8-DOF homography motion admits several particular cases, as discussed in the next sections.

A.2 3 DOFs. Planar motion. Euclidean transformation on the image plane, SE(2). Isometry

The warp of the planer motion [113] is given by (A.1) with

$$\mathtt{H}^E(t_k;\theta) \doteq \begin{pmatrix} \mathtt{R}(t_k\omega_Z) & t_k\mathbf{v} \\ 0^\top & 1 \end{pmatrix}, \tag{A.7}$$

where $\mathbf{v}, \omega_Z$ comprise the 3 DOFs of a translation and an in-plane rotation. The in-plane rotation is

$$\mathtt{R}(\varphi) = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}. \tag{A.8}$$

Divergence. Since

$$\begin{pmatrix} \mathtt{A} & \mathbf{b} \\ 0^\top & 1 \end{pmatrix}^{-1} = \begin{pmatrix} \mathtt{A}^{-1} & -\mathtt{A}^{-1}\mathbf{b} \\ 0^\top & 1 \end{pmatrix} \tag{A.9}$$

and $\mathtt{R}^{-1}(\varphi) = \mathtt{R}(-\varphi)$, we have

$$\begin{pmatrix} \mathbf{x}'_k \\ 1 \end{pmatrix} \sim \begin{pmatrix} \mathtt{R}(-t_k\omega_Z) & -\mathtt{R}(-t_k\omega_Z)(t_k\mathbf{v}) \\ 0^\top & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_k \\ 1 \end{pmatrix}. \tag{A.10}$$

Hence, in Euclidean coordinates the warp is

$$\mathbf{x}'_k = \mathtt{R}(-t_k\omega_Z)(\mathbf{x}_k - t_k\mathbf{v}). \tag{A.11}$$

The flow corresponding to (A.11) is:

$$\mathbf{f} = \frac{\partial \mathbf{x}'}{\partial t} = \mathtt{R}^\top\Big(\frac{\pi}{2} + t\omega_Z\Big)(\mathbf{x} - t\mathbf{v})\omega_Z - \mathtt{R}(-t\omega_Z)\mathbf{v}, \tag{A.12}$$

whose divergence is

$$\nabla \cdot \mathbf{f} = -2\omega_Z \sin(t\omega_Z). \tag{A.13}$$

Hence, for small angles $|t\omega_Z| \ll 1$, the divergence of the flow vanishes.

Deformation. Substituting (A.7) into (A.4),

$$\det(\mathtt{J}_k) = 1, \tag{A.14}$$

since $\det(\mathtt{H}^E) = 1$ and $(\mathbf{e}_3^\top \mathtt{H}^E\, \mathbf{x}_k^b(t))^3 = 1$.

Rate of change of area deformation. Using (A.5) and (A.9), the differential transformation is given by

$$\begin{aligned}
\mathtt{H}^E_{t,t+\Delta t} &= \mathtt{H}^E(t+\Delta t;\theta)\,(\mathtt{H}^E)^{-1}(t;\theta) \\
&= \begin{pmatrix} \mathtt{R}((t+\Delta t)\omega_Z) & (t+\Delta t)\mathbf{v} \\ 0^\top & 1 \end{pmatrix} \begin{pmatrix} \mathtt{R}(-t\omega_Z) & -\mathtt{R}(-t\omega_Z)(t\mathbf{v}) \\ 0^\top & 1 \end{pmatrix} \\
&= \begin{pmatrix} \mathtt{R}((t+\Delta t)\omega_Z)\mathtt{R}(-t\omega_Z) & -\mathtt{R}((t+\Delta t)\omega_Z)\mathtt{R}(-t\omega_Z) + (t+\Delta t)\mathbf{v} \\ 0^\top & 1 \end{pmatrix} \\
&= \begin{pmatrix} \mathtt{R}(\Delta t\omega_Z) & -\mathtt{R}(\Delta t\omega_Z) + (t+\Delta t)\mathbf{v} \\ 0^\top & 1 \end{pmatrix}.
\end{aligned} \tag{A.15}$$

Similarly to (A.14), we obtain $\det(\mathtt{J}_{t,t+\Delta t}) = 1$, since $\det(\mathtt{H}^E_{t,t+\Delta t}) = 1 = (\mathbf{e}_3^\top \mathtt{H}^E_{t,t+\Delta t}\, \mathbf{x}^b(t))^3$.

In short, this warp has the same determinant, rate of change of area deformation, and approximate zero divergence as the 2-DOF feature flow warp (Sec. 3.5.1), which is well-posed. Note, however, that the trajectories are not straight in space-time.

A.3 3 DOFs. Camera rotation, SO(3)

Using calibrated and homogeneous coordinates[47,46], the warp of the 3-DOF rotational motion is given by (A.1) with

$$\mathtt{H}^{O}(t_k;\theta) \doteq \mathtt{R}(t_k\omega) \tag{A.16}$$

where $\theta = \omega = (\omega_1, \omega_2, \omega_3)^\top$ is the angular velocity, and $\mathtt{R}$ (3×3 rotation matrix in space) is parametrized using exponential coordinates (Rodrigues rotation formula[108,48]).

Divergence. As shown in (3.23) in Sec. 3.5.2, the flow for the rotational motion is

$$\nabla \cdot \mathbf{f} = 3(x\omega_2 - y\omega_1). \tag{A.17}$$

Deformation. Substituting (A.16) into (A.4) and letting $\mathbf{r}_{3,k}^\top$ be the third row of $\mathtt{R}(t_k\omega)$ gives (3.24), the area deformation is

$$\det(\mathtt{J}_k) = (\mathbf{r}_{3,k}^\top \mathbf{x}_k^b)^{-3}, \tag{A.18}$$

since $\det(\mathtt{H}^O) = 1$.

Connection between divergence and deformation. If the rotation angle $t_k\|\omega\|$ is small, using the first two terms of the exponential map we approximate $\mathtt{R}(t_k\omega) \approx \mathtt{Id} + (t_k\omega)^\wedge$, where the hat operator $^\wedge$ in $SO(3)$ represents the cross product matrix[7]. Then, $\mathbf{r}_{3,k}^\top \mathbf{x}_k^b \approx (-t_k\omega_2, t_k\omega_1, 1)^\top (x_k, y_k, 1) = 1 + (y_k\omega_1 - x_k\omega_2)t_k$. Substituting this expression into (3.24) and using the first two terms in Taylor's expansion around $z = 0$ of $(1+z)^{-3} \approx 1 - 3z + 6z^2$ (convergent for $|z| < 1$) gives $\det(\mathtt{J}_k) \approx 1 + 3(x_k\omega_2 - y_k\omega_1)t_k$. Notably, the divergence (3.23) and the approximate amplification factor depend linearly on $3(x_k\omega_2 - y_k\omega_1)$. This resemblance is seen in the divergence and deformation maps of the bottom rows in Fig. 3.9 (ECD dataset).

Rate of change of area deformation. Rotation matrices have unit determinant and simplify (A.5) as: $\mathtt{H}^O_{t,t+\Delta t} = \mathtt{R}((t+\Delta t)\omega)\,\mathtt{R}^{-1}(t\omega) = \exp\big(((t+\Delta t)\omega)^\wedge\big)\exp\big((-t\omega)^\wedge\big) = \exp((\Delta t\,\omega)^\wedge) = \mathtt{R}(\omega\Delta t)$. This holds because the rotation axis ω is unique. Hence (A.6) becomes:

$$\det\big(\mathtt{J}_{t,t+\Delta t}\big) = \frac{1}{(\mathbf{e}_3^\top \mathtt{R}(\omega\Delta t)\,\mathbf{x}^b)^3}, \tag{A.19}$$

which is (3.26). Computing the derivative yields (3.27):

$$\begin{aligned}\left.\frac{d|\mathtt{J}_{t,t+\Delta t}|}{d\Delta t}\right|_{\Delta t=0} &= \frac{-3\mathbf{x}^{h\top}(t)}{\left(\mathbf{r}_3^\top(\omega\Delta t)\mathbf{x}^h(t)\right)^4}\left.\frac{d}{d\Delta t}\mathbf{r}_3(\omega\Delta t)\right|_{\Delta t=0} \\ &= 3\mathbf{x}^{h\top}(t)\omega^\wedge\mathbf{e}_3, \end{aligned} \tag{A.20}$$

because

$$\begin{aligned}\frac{d}{d\Delta t}\mathbf{r}_3(\omega\Delta t) &= \frac{d}{d\Delta t}\mathtt{R}^\top(\omega\Delta t)\mathbf{e}_3 \\ &\approx \frac{d}{d\Delta t}\left(\mathtt{Id} - (\omega\Delta t)^\wedge\right)\mathbf{e}_3 \\ &= -\omega^\wedge\mathbf{e}_3 \\ &= (-\omega_y, \omega_x, 0)^\top. \end{aligned} \tag{A.21}$$

A.4 4 DOFs. In-plane camera motion approximation

For completeness, we analyze the warp presented in[99], although we do not particularize (A.1) since it is an approximation. The warp given in[99], which is

$$\mathbf{x}'_k = \mathbf{x}_k - t_k\left(\mathbf{v} + (h_z+1)\mathtt{R}(\varphi)\mathbf{x}_k - \mathbf{x}_k\right), \tag{A.22}$$

has 4 DOFs: $\theta = (\mathbf{v}, \varphi, h_z)^\top$.

Divergence. The flow corresponding to (A.22) is given by

$$\mathbf{f} = \frac{\partial\mathbf{x}'}{\partial t} = -\left(\mathbf{v} + (h_z+1)\mathtt{R}(\varphi)\mathbf{x} - \mathbf{x}\right), \tag{A.23}$$

whose divergence is:

$$\begin{aligned}\nabla\cdot\mathbf{f} &= -(h_z+1)\nabla\cdot\left(\mathtt{R}(\varphi)\mathbf{x}\right) + \nabla\cdot\mathbf{x} && \text{(A.24)} \\ &= 2 - 2(h_z+1)\cos(\varphi). && \text{(A.25)}\end{aligned}$$

Deformation. The Jacobian and its determinant are:

$$\mathtt{J}_k = \frac{\partial\mathbf{x}'_k}{\partial\mathbf{x}_k} = (1+t_k)\mathtt{Id} - (h_z+1)t_k\mathtt{R}(\varphi), \tag{A.26}$$

$$\det(\mathtt{J}_k) = (1+t_k)^2 - 2(1+t_k)t_k(h_z+1)\cos\varphi + t_k^2(h_z+1)^2. \tag{A.27}$$

We skip the rate of change of area deformation since the homography is not given. As particular cases of this warp, one can identify:

- 1-DOF Zoom in/out ($\mathbf{v}=0, \varphi=0$). $\mathbf{x}'_k = (1-t_k h_z)\mathbf{x}_k$.
- 2-DOF translation ($\varphi=0, h_z=0$). $\mathbf{x}'_k = \mathbf{x}_k - t_k\mathbf{v}$.
- 1-DOF "rotation" ($\mathbf{v}=0, h_z=0$). $\mathbf{x}'_k = \mathbf{x}_k - t_k\left(\mathtt{R}(\varphi)\mathbf{x}_k - \mathbf{x}_k\right)$.
  Using a couple of approximations of the exponential map in $SO(2)$, we obtain

$$\begin{aligned}
\mathbf{x}'_k &= \mathbf{x}_k - t_k\left(\mathtt{R}(\varphi) - \mathtt{Id}\right)\mathbf{x}_k && && \text{(A.28)}\\
&\approx \mathbf{x}_k - t_k\varphi^\wedge\mathbf{x}_k && \text{if } \varphi \text{ is small} && \text{(A.29)}\\
&= (\mathtt{Id} + (-t_k\varphi)^\wedge)\mathbf{x}_k && && \text{(A.30)}\\
&\approx \mathtt{R}(-t_k\varphi)\mathbf{x}_k && \text{if } t_k\varphi \text{ is small.} && \text{(A.31)}
\end{aligned}$$

  Hence, φ plays the role of a small angular velocity ω_Z around the camera's optical axis Z, i.e., in-plane rotation.

- 3-DOF planar motion ("isometry") ($h_z = 0$). Using the previous result, the warp splits into translational and rotational components:

$$\begin{aligned}
\mathbf{x}'_k &= \mathbf{x}_k - t_k\left(\mathbf{v} + \mathtt{R}(\varphi)\mathbf{x}_k - \mathbf{x}_k\right) && \text{(A.32)}\\
&\overset{\text{(A.31)}}{\approx} -t_k\mathbf{v} + \mathtt{R}(-t_k\varphi)\mathbf{x}_k. && \text{(A.33)}
\end{aligned}$$

A.5 4 DOFs. Similarity transformation on the image plane. Sim(2)

Another 4-DOF warp is proposed in [113]. Its DOFs are the linear, angular and scaling velocities on the image plane: $\theta = (\mathbf{v}, \omega_Z, s)^\top$. Letting $\beta_k = 1 + t_k s$, the warp is given by (A.1) with

$$\mathtt{H}^S(t_k;\theta) \doteq \begin{pmatrix} \beta_k\mathtt{R}(t_k\omega_Z) & t_k\mathbf{v} \\ 0^\top & 1 \end{pmatrix}. \tag{A.34}$$

Using (A.9) gives

$$\begin{pmatrix} \mathbf{x}'_k \\ 1 \end{pmatrix} \sim \begin{pmatrix} \beta_k^{-1}\mathtt{R}(-t_k\omega_Z) & -\beta_k^{-1}\mathtt{R}(-t_k\omega_Z)(t_k\mathbf{v}) \\ 0^\top & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_k \\ 1 \end{pmatrix}. \tag{A.35}$$

Hence, in Euclidean coordinates the warp is

$$\mathbf{x}'_k = \beta_k^{-1}\mathtt{R}(-t_k\omega_Z)(\mathbf{x}_k - t_k\mathbf{v}). \tag{A.36}$$

Divergence. To compute the flow of (A.36), there are three time-dependent factors. Hence, applying the product rule we obtain three terms, and substituting (A.43) (with $\varphi = -t\omega_Z$) gives:

$$\mathbf{f}_k = \Big(\frac{\partial\beta_k^{-1}}{\partial t_k}\mathtt{R}(-t_k\omega_Z) + \beta_k^{-1}\omega_Z\mathtt{R}^\top\big(\frac{\pi}{2} + t_k\omega_Z\big)\Big)(\mathbf{x}_k - t_k\mathbf{v}) - \beta_k^{-1}\mathtt{R}(-t_k\omega_Z)\mathbf{v}, \tag{A.37}$$

where, by the chain rule,

$$\frac{\partial\beta_k^{-1}}{\partial t_k} = -\beta_k^{-2}\frac{\partial\beta_k}{\partial t_k} = -\beta_k^{-2}s = -\frac{s}{(1+t_ks)^2}. \tag{A.38}$$

Hence, the divergence of the flow is:

$$\nabla\cdot\mathbf{f}_k = \frac{\partial\beta_k^{-1}}{\partial t_k}\nabla\cdot\Big(\mathtt{R}(-t_k\omega_Z)\mathbf{x}_k\Big) + \beta_k^{-1}\omega_Z\nabla\cdot\Big(\mathtt{R}^\top\big(\frac{\pi}{2} + t_k\omega_Z\big)\mathbf{x}_k\Big) \tag{A.39}$$

$$= \frac{\partial\beta_k^{-1}}{\partial t_k}2\cos(t_k\omega_Z) + \beta_k^{-1}\omega_Z 2\sin(-t_k\omega_Z) \tag{A.40}$$

The formulas for $SE(2)$ are obtained from the above ones with $s = 0$ (i.e., $\beta_k = 1$).

Deformation. The Jacobian and its determinant are:

$$\mathtt{J}_k = \frac{\partial\mathbf{x}'_k}{\partial\mathbf{x}_k} = \beta_k^{-1}\mathtt{R}(-t_k\omega_Z), \tag{A.41}$$

$$\det(\mathtt{J}_k) = \beta_k^{-2} = \frac{1}{(1+t_ks)^2}. \tag{A.42}$$

The following result will be useful to simplify equations. For a 2D rotation $\mathtt{R}(\varphi(t))$, it holds that:

$$\frac{\partial\mathtt{R}(\varphi(t))}{\partial t} = -\mathtt{R}^\top\Big(\frac{\pi}{2} - \varphi\Big)\frac{\partial\varphi}{\partial t}. \tag{A.43}$$

Rate of change of area deformation. Using (A.5) it gives $\mathtt{H}_{t,t+\Delta t} = \mathtt{H}^S(t+\Delta t;\theta)\,(\mathtt{H}^S)^{-1}(t;\theta)$. Similarities form a matrix Lie group, hence the inverse and the product of two similarities is

also a similarity. Since $\mathtt{H}_{t,t+\Delta t}$ is a similarity, its third row is $\mathbf{e}_3^\top \mathtt{H}_{t,t+\Delta t} = (0,0,1)$, which makes the denominator in (A.4) equal to one. Substituting in (A.4) produces

$$\begin{aligned} |\mathtt{J}_{t,t+\Delta t}| &= \left|\det\left(\mathtt{H}^S(t+\Delta t;\theta)\,(\mathtt{H}^S)^{-1}(t;\theta)\right)\right| \\ &= \left|\frac{\det\left(\mathtt{H}^S(t+\Delta t;\theta)\right)}{\det\left(\mathtt{H}^S(t;\theta)\right)}\right| \\ &= \left(\frac{\beta(t+\Delta t;s)}{\beta(t;s)}\right)^2. \end{aligned} \tag{A.44}$$

Two intuitive remarks about (A.44): (*i*) it only depends on the scaling DOF (i.e., one out of the four DOFs) since ω_z and $\mathbf{v}$ do not appear; and (*ii*) it can be used to derive the 1-DOF formula (3.18): the 1-DOF scaling transformation is modeled by $\mathtt{H}^S$ with $\omega_z = 0$, $\mathbf{v} = 0$ and $\beta(t;h_z) = (1-th_z)^{-1}$. Substituting this choice of β in (A.44) makes it coincide with (3.18).

The 4-DOF transformation in [99] has a similar geometric meaning but a different parametrization. Hence, we use the above result and penalize collapse by means of the corresponding scaling parameter in [99].

A.6 6 DOFs. Affine Transformation on the Image Plane

A planar affine transformation has 6 DOFs in θ. Letting

$$\mathtt{H}^A(t;\theta) \doteq \begin{pmatrix} \mathtt{A}(t;\theta) & t\mathbf{b} \\ 0^\top & 1 \end{pmatrix}, \tag{A.45}$$

using (A.9) gives

$$\begin{pmatrix} \mathbf{x}'_k \\ 1 \end{pmatrix} \sim \begin{pmatrix} \mathtt{A}^{-1}(t_k;\theta) & -\mathtt{A}^{-1}(t_k;\theta)(t_k\mathbf{b}) \\ 0^\top & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x}_k \\ 1 \end{pmatrix}. \tag{A.46}$$

Hence, the warp in Euclidean coordinates is

$$\mathbf{x}'_k = \mathtt{A}^{-1}(t_k;\theta)(\mathbf{x}_k - t_k\mathbf{b}). \tag{A.47}$$

Divergence. The flow corresponding to (A.47) is given by

$$\mathbf{f} = \frac{\partial \mathbf{x}'}{\partial t} = \frac{\partial \mathtt{A}^{-1}(t;\theta)}{\partial t}(\mathbf{x} - t\mathbf{b}) - \mathtt{A}^{-1}(t;\theta)\mathbf{b}, \tag{A.48}$$

whose divergence is:

$$\nabla \cdot \mathbf{f} = \nabla \cdot \Big(\frac{\partial \mathtt{A}^{-1}(t;\theta)}{\partial t}(\mathbf{x} - t\mathbf{b}) - \mathtt{A}^{-1}(t;\theta)\mathbf{b} \Big) \tag{A.49}$$

$$= \nabla \cdot \Big(\frac{\partial \mathtt{A}^{-1}(t;\theta)}{\partial t}\mathbf{x} \Big) \tag{A.50}$$

$$= \mathrm{tr}\Big(\frac{\partial \mathtt{A}^{-1}(t;\theta)}{\partial t} \Big). \tag{A.51}$$

Deformation. Substituting (A.45) into (A.4), the area deformation is

$$\det(\mathtt{J}_k) = \det(\mathtt{H}^A(t_k;\theta)) = \det(\mathtt{A}(t;\theta)), \tag{A.52}$$

since $\mathbf{e}_3^\top \mathtt{H}^A = (0,0,1)$ and the denominator in (A.4) becomes one.

Rate of change of area deformation. Now the incremental change (A.5) gives $\mathtt{H}_{t,t+\Delta t} = \mathtt{H}^A(t+\Delta t;\theta)\,(\mathtt{H}^A)^{-1}(t;\theta)$. Affinities also form a matrix Lie group, hence $\mathtt{H}_{t,t+\Delta t}$ is an affinity. Using (A.52), following similar steps as those in (A.44) yields

$$\begin{aligned} |\mathtt{J}_{t,t+\Delta t}| &= \Big|\det\big(\mathtt{H}^A(t+\Delta t;\theta)\,(\mathtt{H}^A)^{-1}(t;\theta)\big)\Big| \\ &= \left|\frac{\det\big(\mathtt{A}(t+\Delta t;\theta)\big)}{\det\big(\mathtt{A}(t;\theta)\big)}\right|. \end{aligned} \tag{A.53}$$

Notice that the 2×2 matrix $\mathtt{A}$ includes not only a scaling parameter but also a shear, which affects the area deformation.

B

Solutions for Space-time Flow

B.1 Time-Awareness: PDE solutions

The proposed *time-aware flow* in Sec. 4.2.3 is given as the solution to (4.7). Letting the flow be $\mathbf{v} = (v_x, v_y)^\top$, the system of PDEs can be written as:

$$\begin{aligned} v_x \frac{\partial v_x}{\partial x} + v_y \frac{\partial v_x}{\partial y} + \frac{\partial v_x}{\partial t} = 0, \\ v_x \frac{\partial v_y}{\partial x} + v_y \frac{\partial v_y}{\partial y} + \frac{\partial v_y}{\partial t} = 0. \end{aligned} \tag{B.1}$$

Upwind and Burgers' schemes can be used to discretize and numerically solve the system of PDEs[38,141].

Discretization. Let $\mathbf{v}^n(x, y)$ be the flow vector at discretized space- (e.g., pixel) and time-indices (x, y, n), with discretization steps Δx, Δy, and Δt, respectively, and let the forward

(+) and backward (−) differences of a scalar field w (e.g., v_x^n or v_y^n) be defined as

$$\begin{aligned} D_x^+ w \equiv \frac{\partial w}{\partial x}^+ &= \frac{1}{\Delta x}\big(w(x+\Delta x, y) - w(x,y)\big), \\ D_y^+ w \equiv \frac{\partial w}{\partial y}^+ &= \frac{1}{\Delta y}\big(w(x, y+\Delta y) - w(x,y)\big), \end{aligned} \tag{B.2}$$

and

$$\begin{aligned} D_x^- w \equiv \frac{\partial w}{\partial x}^- &= \frac{1}{\Delta x}\big(w(x,y) - w(x-\Delta x, y)\big), \\ D_y^- w \equiv \frac{\partial w}{\partial y}^- &= \frac{1}{\Delta y}\big(w(x,y) - w(x, y-\Delta y)\big). \end{aligned} \tag{B.3}$$

We discretize in time using forward differences, $\frac{\partial w}{\partial t} \approx (w(t+\Delta t) - w(t))/\Delta t$, to yield explicit update schemes: $w(t+\Delta t) \approx w(t) + \Delta t \frac{\partial w}{\partial t}$.

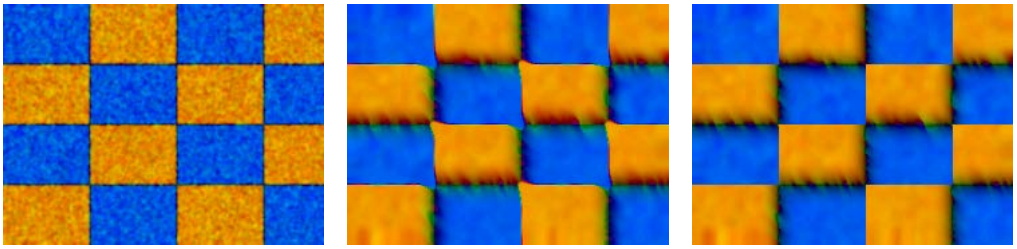
B.2 Upwind scheme

The first-order upwind scheme is an explicit scheme that updates the flow as follows, based on the sign of the variables: it uses $D_x^+ v_x^n$ and $D_x^+ v_y^n$ for $v_x^n > 0$ ($D_x^- v_x^n$ and $D_x^- v_y^n$ otherwise), and $D_y^+ v_x^n$ and $D_y^+ v_y^n$ for $v_y^n > 0$ ($D_y^- v_x^n$ and $D_y^- v_y^n$ otherwise). The scheme is stable if the flow satisfies $\Delta t \max\{|v_x|/\Delta x + |v_y|/\Delta y\} < 1$ (CFL stability condition[72]). For example, in case that $v_x^n > 0$ and $v_y^n > 0$ at the current discretization time n:

$$\begin{aligned} v_x^{n+1} &= v_x^n - \Delta t \left(v_x^n D_x^+ v_x^n + v_y^n D_y^+ v_x^n \right), \\ v_y^{n+1} &= v_y^n - \Delta t \left(v_y^n D_y^+ v_y^n + v_x^n D_x^+ v_y^n \right). \end{aligned} \tag{B.4}$$

B.3 Burgers' scheme

The study of the inviscid Burgers' equation provides a more conservative scheme solution, especially at "shock" and "fan wave" cases[141]. In this explicit scheme, the product terms in the same variable (which convey that the flow is transporting itself), $v_x^n D_x^+ v_x^n$ and $v_y^n D_y^+ v_y^n$ in



(a) Original flow **(b)** Upwind scheme **(c)** Burgers' scheme

Figure B.1: *Comparison of the two flow propagation schemes.* Original flow (a) has large shock and fan waves (the color changes between orange and blue) to highlight the difference. The propagated flows with both schemes are shown in (b) (c). Same color notation as Fig. 1.1.

(B.4), are replaced with U_x and U_y respectively, which are given by:

$$
\begin{aligned}
U_x &= \frac{1}{2}\left(\operatorname{sgn}\left(v_x^n(x,y)\right)\left(v_x^n(x,y)\right)^2 + F_x - B_x\right), \\
F_x &= \begin{cases} \left(v_x^n(x+\Delta x,y)\right)^2, & \text{if } v_x^n(x+\Delta x,y) < 0 \\ 0, & \text{otherwise} \end{cases} \\
B_x &= \begin{cases} \left(v_x^n(x-\Delta x,y)\right)^2, & \text{if } v_x^n(x-\Delta x,y) > 0 \\ 0, & \text{otherwise} \end{cases}
\end{aligned}
\tag{B.5}
$$

and

$$
\begin{aligned}
U_y &= \frac{1}{2}\left(\operatorname{sgn}\left(v_y^n(x,y)\right)\left(v_y^n(x,y)\right)^2 + F_y - B_y\right), \\
F_y &= \begin{cases} \left(v_y^n(x,y+\Delta y)\right)^2, & \text{if } v_y^n(x,y+\Delta y) < 0 \\ 0, & \text{otherwise} \end{cases} \\
B_y &= \begin{cases} \left(v_y^n(x,y-\Delta y)\right)^2, & \text{if } v_y^n(x,y-\Delta y) > 0 \\ 0. & \text{otherwise} \end{cases}
\end{aligned}
\tag{B.6}
$$

B.4 Comparison of the schemes

Figure B.1 shows the comparison between the two schemes, especially for the "shock" and "fan wave" cases. After some iterations of the propagation, the upwind scheme starts to produce artifacts around the shock and fan flows (the color boundary of orange and blue), while the Burgers' scheme provides more stable flows.

References

[1] Akolkar, H., Ieng, S.-H., & Benosman, R. (2022). Real-time high speed motion prediction using fast aperture-robust event-driven visual flow. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(1), 361–372.

[2] Almatrafi, M., Baldwin, R., Aizawa, K., & Hirakawa, K. (2020). Distance surface for event-based optical flow. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(7), 1547–1556.

[3] Angelopoulos, A. N., Martel, J. N., Kohli, A. P., Conradt, J., & Wetzstein, G. (2020). Event based, near eye gaze tracking beyond 10,000 hz. *IEEE Trans. Vis. Comput. Graphics*.

[4] Atcheson, B., Heidrich, W., & Ihrke, I. (2009). An evaluation of optical flow algorithms for background oriented schlieren imaging. *Experiments in fluids*, 46, 467–476.

[5] Baker, S. & Matthews, I. (2004). Lucas-kanade 20 years on: A unifying framework. *Int. J. Comput. Vis.*, 56(3), 221–255.

[6] Bardow, P., Davison, A. J., & Leutenegger, S. (2016). Simultaneous optical flow and intensity estimation from an event camera. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 884–892).

[7] Barfoot, T. D. (2015). *State Estimation for Robotics - A Matrix Lie Group Approach*. Cambridge University Press.

[8] Barkas, S. N. (2005). An introduction to fast poisson solvers. *Philips J Res*, 37(5-6), 231–264.

[9] Barlow, H. B. & Levick, W. R. (1965). The mechanism of directionally selective units in rabbit's retina. *J. Physiol.*, 178(3), 477–504.

[10] Benosman, R., Clercq, C., Lagorce, X., Ieng, S.-H., & Bartolozzi, C. (2014). Event-based visual flow. *IEEE Trans. Neural Netw. Learn. Syst.*, 25(2), 407–417.

[11] Benosman, R., Ieng, S.-H., Clercq, C., Bartolozzi, C., & Srinivasan, M. (2012). Asynchronous frameless event-based optical flow. *Neural Netw.*, 27, 32–37.

[12] Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 24.

[13] Brandli, C., Berner, R., Yang, M., Liu, S.-C., & Delbruck, T. (2014a). A 240x180 130dB 3μs latency global shutter spatiotemporal vision sensor. *IEEE J. Solid-State Circuits*, 49(10), 2333–2341.

[14] Brandli, C., Muller, L., & Delbruck, T. (2014b). Real-time, high-speed video decompression using a frame- and event-based DAVIS sensor. In *IEEE Int. Symp. Circuits Syst. (ISCAS)* (pp. 686–689).

[15] Brebion, V., Moreau, J., & Davoine, F. (2021). Real-time optical flow for vehicular perception with low- and high-resolution event cameras. *IEEE Trans. Intell. Transportation Systems*, (pp. 1–13).

[16] Brosch, T., Tschechne, S., & Neumann, H. (2015). On event-based optical flow detection. *Front. Neurosci.*, 9(137).

[17] Bryner, S., Gallego, G., Rebecq, H., & Scaramuzza, D. (2019). Event-based, direct camera tracking from a photometric 3D map using nonlinear optimization. In *IEEE Int. Conf. Robot. Autom. (ICRA)*.

[18] Cannici, M., Ciccone, M., Romanoni, A., & Matteucci, M. (2020). A differentiable recurrent surface for asynchronous event-based data. In *Eur. Conf. Comput. Vis. (ECCV)* (pp. 136–152).

[19] Charbonnier, P., Blanc-Feraud, L., Aubert, G., & Barlaud, M. (1997). Deterministic edge-preserving regularization in computed imaging. *IEEE Trans. Image Process.*, 6(2), 298–311.

[20] Chin, T., Bagchi, S., Eriksson, A. P., & van Schaik, A. (2019). Star tracking using an event camera. In *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)* (pp. 1646–1655).

[21] Clady, X., Clercq, C., Ieng, S.-H., Houseini, F., Randazzo, M., Natale, L., Bartolozzi, C., & Benosman, R. (2014). Asynchronous visual event-based time-to-contact. *Front. Neurosci.*, 8(9).

[22] Clady, X., Ieng, S.-H., & Benosman, R. (2015). Asynchronous event-based corner detection and matching. *Neural Netw.*, 66, 91–106.

[23] Clark, D. A. & Demb, J. B. (2016). Parallel computations in insect and mammalian visual motion processing. *Current Biology*, 26(20), R1062–R1072.

[24] Cohen, G., Afshar, S., & van Schaik, A. (2018). Approaches for astrometry using event-based sensors. In *Proc. Advanced Maui Optical and Space Surveillance Technol. Conf. (AMOS)*.

[25] Cohen, G., Afshar, S., van Schaik, A., Wabnitz, A., Bessell, T., Rutten, M., & Morreale, B. (2017). Event-based sensing for space situational awareness. In *Proc. Advanced Maui Optical and Space Surveillance Technol. Conf. (AMOS)*.

[26] Cook, M., Gugelmann, L., Jug, F., Krautz, C., & Steger, A. (2011). Interacting maps for fast visual interpretation. In *Int. Joint Conf. Neural Netw. (IJCNN)* (pp. 770–776).

[27] Corke, P. (2017). *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer Tracts in Advanced Robotics. Springer.

[28] D'Angelo, G., Janotte, E., Schoepe, T., O'Keeffe, J., Milde, M. B., Chicca, E., & Bartolozzi, C. (2020). Event-based eccentric motion detection exploiting time difference encoding. *Front. Neurosci.*, 14, 451.

[29] Dardelet, L., Benosman, R., & Ieng, S.-H. (2021). An event-by-event feature detection and tracking invariant to motion direction and velocity. *TechRxiv preprint*.

[30] Delbruck, T. (2008). Frame-free dynamic digital vision. In *Proc. Int. Symp. Secure-Life Electron.* (pp. 21–26).

[31] Delmerico, J., Cieslewski, T., Rebecq, H., Faessler, M., & Scaramuzza, D. (2019). Are we ready for autonomous drone racing? the UZH-FPV drone racing dataset. In *IEEE Int. Conf. Robot. Autom. (ICRA)* (pp. 6713–6719).

[32] Dimitrova, R. S., Gehrig, M., Brescianini, D., & Scaramuzza, D. (2020). Towards low-latency high-bandwidth control of quadrotors using event cameras. In *IEEE Int. Conf. Robot. Autom. (ICRA)* (pp. 4294–4300).

[33] Ding, Y., Li, Z., Chen, Z., Ji, Y., Yu, J., & Ye, J. (2023). Full-volume 3d fluid flow reconstruction with light field piv. *IEEE Trans. Pattern Anal. Mach. Intell.*

[34] Ding, Z., Zhao, R., Zhang, J., Gao, T., Xiong, R., Yu, Z., & Huang, T. (2022). Spatio-temporal recurrent networks for event-based optical flow estimation. *AAAI Conf. Artificial Intell.*, 36(1), 525–533.

[35] Drazen, D., Lichtsteiner, P., Häfliger, P., Delbrück, T., & Jensen, A. (2011). Toward real-time particle tracking using an event-based dynamic vision sensor. *Experim. Fluids*, 51(5), 1465–1469.

[36] Duan, P., Wang, Z., Shi, B., Cossairt, O., Huang, T., & Katsaggelos, A. (2021a). Guided event filtering: Synergy between intensity images and neuromorphic events for high performance imaging. *IEEE Trans. Pattern Anal. Mach. Intell.*, (pp. 1–1).

[37] Duan, P., Wang, Z. W., Zhou, X., Ma, Y., & Shi, B. (2021b). Eventzoom: Learning to denoise and super resolve neuromorphic events. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 12824–12833).

[38] Evans, L. C. (2010). *Partial Differential Equations*. Graduate Studies in Mathematics. American Mathematical Society.

[39] Falanga, D., Kleber, K., & Scaramuzza, D. (2020). Dynamic obstacle avoidance for quadrotors with event cameras. *Science Robotics*, 5(40), eaaz9712.

[40] Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. In *Scandinavian Conf. on Im. Analysis (SCIA)* (pp. 363–370).

[41] Finateu, T., Niwa, A., Matolin, D., Tsuchimoto, K., Mascheroni, A., Reynaud, E., Mostafalu, P., Brady, F., Chotard, L., LeGoff, F., Takahashi, H., Wakabayashi, H., Oike, Y., & Posch, C. (2020). A 1280x720 back-illuminated stacked temporal contrast event-based vision sensor with 4.86μm pixels, 1.066Geps readout, programmable event-rate controller and compressive data-formatting pipeline. In *IEEE Intl. Solid-State Circuits Conf. (ISSCC)* (pp. 112–114).

[42] Fitzgerald, J. E. & Clark, D. A. (2015). Nonlinear circuits for naturalistic visual motion estimation. *Elife*, 4, e09123.

[43] Gallego, G., Delbruck, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., Leutenegger, S., Davison, A., Conradt, J., Daniilidis, K., & Scaramuzza, D. (2022). Event-based vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(1), 154–180.

[44] Gallego, G., Forster, C., Mueggler, E., & Scaramuzza, D. (2015). Event-based camera pose tracking using a generative event model. arXiv:1510.01972.

[45] Gallego, G., Gehrig, M., & Scaramuzza, D. (2019). Focus is all you need: Loss functions for event-based vision. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 12272–12281).

[46] Gallego, G., Rebecq, H., & Scaramuzza, D. (2018). A unifying contrast maximization framework for event cameras, with applications to motion, depth, and optical flow estimation. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 3867–3876).

[47] Gallego, G. & Scaramuzza, D. (2017). Accurate angular velocity estimation with an event camera. *IEEE Robot. Autom. Lett.*, 2(2), 632–639.

[48] Gallego, G. & Yezzi, A. (2014). A compact formula for the derivative of a 3-D rotation in exponential coordinates. *J. Math. Imaging Vis.*, 51(3), 378–384.

[49] Gallego, G., Yezzi, A., Fedele, F., & Benetazzo, A. (2011). A variational stereo method for the three-dimensional reconstruction of ocean waves. *IEEE Trans. Geosci. Remote Sens.*, 49(11), 4445–4457.

[50] Gao, Y., Li, S., Li, Y., Guo, Y., & Dai, Q. (2022). Superfast: 200× video frame interpolation via event camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, (pp. 1–17).

[51] Gehrig, D., Gehrig, M., Hidalgo-Carrió, J., & Scaramuzza, D. (2020a). Video to Events: Recycling video datasets for event cameras. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*.

[52] Gehrig, D., Loquercio, A., Derpanis, K. G., & Scaramuzza, D. (2019). End-to-end learning of representations for asynchronous event-based data. In *Int. Conf. Comput. Vis. (ICCV)* (pp. 5632–5642).

[53] Gehrig, D., Rebecq, H., Gallego, G., & Scaramuzza, D. (2018). Asynchronous, photometric feature tracking using events and frames. In *Eur. Conf. Comput. Vis. (ECCV)* (pp. 766–781).

[54] Gehrig, D., Rebecq, H., Gallego, G., & Scaramuzza, D. (2020b). EKLT: Asynchronous photometric feature tracking using events and frames. *Int. J. Comput. Vis.*, 128, 601–618.

[55] Gehrig, M., Aarents, W., Gehrig, D., & Scaramuzza, D. (2021a). DSEC: A stereo event camera dataset for driving scenarios. *IEEE Robot. Autom. Lett.*, 6(3), 4947–4954.

[56] Gehrig, M., Millhäusler, M., Gehrig, D., & Scaramuzza, D. (2021b). E-RAFT: Dense optical flow from event cameras. In *Int. Conf. 3D Vision (3DV)* (pp. 197–206).

[57] Gehrig, M., Muglikar, M., & Scaramuzza, D. (2022). Dense continuous-time optical flow from events and frames. *arXiv preprint arXiv:2203.13674*.

[58] Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *Int. J. Robot. Research*, 32(11), 1231–1237.

[59] Glover, A. & Bartolozzi, C. (2016). Event-driven ball detection and gaze fixation in clutter. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)* (pp. 2203–2208).

[60] Goldhahn, E. & Seume, J. (2007). The background oriented schlieren technique: sensitivity, accuracy, resolution and application to a three-dimensional density field. *Experiments in fluids*, 43, 241–249.

[61] Gonzalez, R. C. & Woods, R. E. (2009). *Digital Image Processing*. Pearson Education.

[62] Graca, R. & Delbruck, T. (2021). Unraveling the paradox of intensity-dependent DVS pixel noise. In *Int. Image Sensor Workshop (IISW)*.

[63] Gu, C., Learned-Miller, E., Sheldon, D., Gallego, G., & Bideau, P. (2021). The spatio-temporal Poisson point process: A simple model for the alignment of event camera data. In *Int. Conf. Comput. Vis. (ICCV)* (pp. 13495–13504).

[64] Haessig, G., Cassidy, A., Alvarez-Icaza, R., Benosman, R., & Orchard, G. (2018). Spiking optical flow for event-based sensors using IBM's truenorth neurosynaptic system. *IEEE Trans. Biomed. Circuits Syst.*, 12(4), 860–870.

[65] Hagenaars, J. J., Paredes-Valles, F., & de Croon, G. C. H. E. (2021). Self-supervised learning of event-based optical flow with spiking neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34 (pp. 7167–7179).

[66] Hamann, F. & Gallego, G. (2022). Stereo co-capture system for recording and tracking fish with frame- and event cameras. In *26th International Conference on Pattern Recognition (ICPR), Visual observation and analysis of Vertebrate And Insect Behavior (VAIB) Workshop*.

[67] Hargather, M. J. & Settles, G. S. (2010). Natural-background-oriented schlieren imaging. *Experiments in fluids*, 48(1), 59–68.

[68] Hassenstein, B. & Reichardt, W. (1956). Systemtheoretische analyse der zeit-, reihenfolgen- und vorzeichenauswertung bei der bewegungsperzeption des rüsselkäfers chlorophanus. *Zeitschrift für Naturforschung B*, 11(9-10), 513–524.

[69] Hayasaka, K., Tagawa, Y., Liu, T., & Kameda, M. (2016). Optical-flow-based background-oriented schlieren technique for measuring a laser-induced underwater shock wave. *Experiments in Fluids*, 57, 1–11.

[70] Heineck, J. T., Banks, D. W., Smith, N. T., Schairer, E. T., Bean, P. S., & Robillos, T. (2021). Background-oriented schlieren imaging of supersonic aircraft in flight. *AIAA Journal*, 59(1), 11–21.

[71] Hidalgo-Carrió, J., Gallego, G., & Scaramuzza, D. (2022). Event-aided direct sparse odometry. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 5781–5790).

[72] Hirsch, C. (2007). The resolution of numerical schemes. In C. Hirsch (Ed.), *Numerical Computation of Internal and External Flows* (pp. 411–412). Oxford: Butterworth-Heinemann, 2 edition.

[73] Hooke, R. (1665). Of a new property in the air. *Micrographia, Observation LVIII*, (pp. 217–219).

[74] Huang, Z., Shi, X., Zhang, C., Wang, Q., Cheung, K. C., Qin, H., Dai, J., & Li, H. (2022a). FlowFormer: A transformer architecture for optical flow. In *Eur. Conf. Comput. Vis. (ECCV)* (pp. 668–685).

[75] Huang, Z., Zhang, T., Heng, W., Shi, B., & Zhou, S. (2022b). Real-time intermediate flow estimation for video frame interpolation. In *Eur. Conf. Comput. Vis. (ECCV)* (pp. 624–642).

[76] Javier Hidalgo-Carrio, D. G. & Scaramuzza, D. (2020). Learning monocular dense depth from events. In *Int. Conf. 3D Vision (3DV)*.

[77] Jia, D., Wang, K., Luo, S., Liu, T., & Liu, Y. (2021). Braft: Recurrent all-pairs field transforms for optical flow based on correlation blocks. *IEEE Signal Processing Letters*, 28, 1575–1579.

[78] Jiang, Z., Zhang, Y., Zou, D., Ren, J., Lv, J., & Liu, Y. (2020). Learning event-based motion deblurring. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 3320–3329).

[79] Kim, H., Handa, A., Benosman, R., Ieng, S.-H., & Davison, A. J. (2014). Simultaneous mosaicing and tracking with an event camera. In *British Mach. Vis. Conf. (BMVC)*.

[80] Kim, H. & Kim, H. J. (2021). Real-time rotational motion estimation with contrast maximization over globally aligned events. *IEEE Robot. Autom. Lett.*, 6(3), 6016–6023.

[81] Kim, H., Leutenegger, S., & Davison, A. J. (2016). Real-time 3D reconstruction and 6-DoF tracking with an event camera. In *Eur. Conf. Comput. Vis. (ECCV)* (pp. 349–364).

[82] Kim, J., Bae, J., Park, G., Zhang, D., & Kim, Y. M. (2021). N-imagenet: Towards robust, fine-grained object recognition with event cameras. *Int. Conf. Comput. Vis. (ICCV)*, (pp. 2146–2156).

[83] Kingma, D. P. & Ba, J. L. (2015). Adam: A method for stochastic optimization. *Int. Conf. Learn. Representations (ICLR)*.

[84] Krehl, P. & Engemann, S. (1995). August toepler—the first who visualized shock waves. *Shock Waves*, 5, 1–18.

[85] Kueng, B., Mueggler, E., Gallego, G., & Scaramuzza, D. (2016). Low-latency visual odometry using event-based feature tracks. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)* (pp. 16–23).

[86] Lagorce, X., Orchard, G., Gallupi, F., Shi, B. E., & Benosman, R. (2017). HOTS: A hierarchy of event-based time-surfaces for pattern recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(7), 1346–1359.

[87] Lee, C., Kosta, A., Zhu, A. Z., Chaney, K., Daniilidis, K., & Roy, K. (2020). Spike-flownet: Event-based optical flow estimation with energy-efficient hybrid neural networks. In *Eur. Conf. Comput. Vis. (ECCV)* (pp. 366–382).

[88] Lee, J., Delbruck, T., Park, P. K. J., Pfeiffer, M., Shin, C.-W., Ryu, H., & Kang, B. C. (2012). Live demonstration: Gesture-based remote control using stereo pair of dynamic vision sensors. In *IEEE Int. Symp. Circuits Syst. (ISCAS)*.

[89] Li, H., Li, G., & Shi, L. (2016). Classification of spatiotemporal events based on random forest. In *Advances in Brain Inspired Cognitive Systems (BICS)*.

[90] Li, H., Li, G., & Shi, L. (2019). Super-resolution of spatiotemporal event-stream image. *Neurocomputing*, 335, 206–214.

[91] Lichtsteiner, P., Posch, C., & Delbruck, T. (2008). A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. *IEEE J. Solid-State Circuits*, 43(2), 566–576.

[92] Lin, S., Zhang, J., Pan, J., Jiang, Z., Zou, D., Wang, Y., Chen, J., & Ren, J. (2020). Learning event-driven video deblurring and interpolation. In *Eur. Conf. Comput. Vis. (ECCV)* (pp. 695–710).

[93] Liu, D., Parra, A., & Chin, T.-J. (2020). Globally optimal contrast maximisation for event-based motion estimation. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 6348–6357).

[94] Liu, D., Parra, A., & Chin, T.-J. (2021). Spatiotemporal registration for event-based visual odometry. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 4937–4946).

[95] Liu, H., Moeys, D. P., Das, G., Neil, D., Liu, S.-C., & Delbruck, T. (2016). Combined frame- and event-based detection and tracking. In *IEEE Int. Symp. Circuits Syst. (ISCAS)* (pp. 2511–2514).

[96] Liu, M. & Delbruck, T. (2018). Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. In *British Mach. Vis. Conf. (BMVC)* (pp. 1–12).

[97] Lu, X., Zhou, Y., & Shen, S. (2021). Event-based motion segmentation by cascaded two-level multi-model fitting. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)* (pp. 4445–4452).

[98] Lucas, B. D. & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Int. Joint Conf. Artificial Intell. (IJCAI)* (pp. 674–679).

[99] Mitrokhin, A., Fermuller, C., Parameshwara, C., & Aloimonos, Y. (2018). Event-based moving object detection and tracking. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)* (pp. 1–9).

[100] Mostafavi I., S., Wang, L., & Yoon, K.-J. Y. (2021). Learning to reconstruct hdr images from events, with applications to depth and flow prediction. *Int. J. Comput. Vis.*, 129(4), 900–920.

[101] Mostafavi I., S. M., Choi, J., & Yoon, K.-J. (2020). Learning to super resolve intensity images from events. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 2768–2786).

[102] Mueggler, E., Bartolozzi, C., & Scaramuzza, D. (2017a). Fast event-based corner detection. In *British Mach. Vis. Conf. (BMVC)*.

[103] Mueggler, E., Gallego, G., Rebecq, H., & Scaramuzza, D. (2018). Continuous-time visual-inertial odometry for event cameras. *IEEE Trans. Robot.*, 34(6), 1425–1440.

[104] Mueggler, E., Gallego, G., & Scaramuzza, D. (2015). Continuous-time trajectory estimation for event-based vision sensors. In *Robotics: Science and Systems (RSS)*.

[105] Mueggler, E., Rebecq, H., Gallego, G., Delbruck, T., & Scaramuzza, D. (2017b). The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *Int. J. Robot. Research*, 36(2), 142–149.

[106] Muglikar, M., Gallego, G., & Scaramuzza, D. (2021a). ESL: Event-based structure light. In *Int. Conf. 3D Vision (3DV)* (pp. 1165–1174).

[107] Muglikar, M., Gehrig, M., Gehrig, D., & Scaramuzza, D. (2021b). How to calibrate your event camera. In *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*.

[108] Murray, R. M., Li, Z., & Sastry, S. (1994). *A Mathematical Introduction to Robotic Manipulation*. CRC Press.

[109] Nagata, J., Sekikawa, Y., & Aoki, Y. (2021). Optical flow estimation by matching time surface with event-based cameras. *Sensors*, 21(4).

[110] Ng, M., Er, Z. M., Soh, G. S., & Foong, S. (2022). Aggregation functions for simultaneous attitude and image estimation with event cameras at high angular rates. *IEEE Robot. Autom. Lett.*, (pp. 1–1).

[111] Nguyen, A., Do, T., Caldwell, D. G., & Tsagarakis, N. G. (2019). Real-time 6DOF pose relocalization for event cameras with stacked spatial LSTM networks. In *IEEE Conf. Comput. Vis. Pattern Recog. Workshops (CVPRW)*.

[112] Nunes, U. M. & Demiris, Y. (2020). Entropy minimisation framework for event-based vision model estimation. In *Eur. Conf. Comput. Vis. (ECCV)* (pp. 161–176).

[113] Nunes, U. M. & Demiris, Y. (2021). Robust event-based vision model estimation by dispersion minimisation. *IEEE Trans. Pattern Anal. Mach. Intell.*

[114] Orchard, G., Benosman, R., Etienne-Cummings, R., & Thakor, N. V. (2013). A spiking neural network architecture for visual motion estimation. In *IEEE Biomed. Circuits Syst. Conf. (BioCAS)* (pp. 298–301).

[115] Orchard, G., Jayawant, A., Cohen, G. K., & Thakor, N. (2015). Converting static image datasets to spiking neuromorphic datasets using saccades. *Front. Neurosci.*, 9, 437.

[116] Pan, L., Liu, M., & Hartley, R. (2020). Single image optical flow estimation with an event camera. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 1669–1678).

[117] Parameshwara, C. M., Sanket, N. J., Singh, C. D., Fermüller, C., & Aloimonos, Y. (2021). 0-MMS: Zero-shot multi-motion segmentation with a monocular event camera. In *IEEE Int. Conf. Robot. Autom. (ICRA)* (pp. 9594–9600).

[118] Paredes-Valles, F. & de Croon, G. C. H. E. (2021). Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 3445–3454).

[119] Paredes-Vallés, F., Scheper, K. Y., De Wagter, C., & de Croon, G. C. (2023). Taming contrast maximization for learning sequential, low-latency, event-based optical flow. *arXiv preprint arXiv:2303.05214*.

[120] Paredes-Valles, F., Scheper, K. Y. W., & de Croon, G. C. H. E. (2019). Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception. *IEEE Trans. Pattern Anal. Mach. Intell.*

[121] Peng, X., Gao, L., Wang, Y., & Kneip, L. (2022). Globally-optimal contrast maximisation for event cameras. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(7), 3479–3495.

[122] Peng, X., Wang, Y., Gao, L., & Kneip, L. (2020). Globally-optimal event camera motion estimation. In *Eur. Conf. Comput. Vis. (ECCV)* (pp. 51–67).

[123] Pfrommer, B. (2022). Frequency Cam: Imaging periodic signals in real-time. In *arXiv*.

[124] Posch, C., Matolin, D., & Wohlgenannt, R. (2011). A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS. *IEEE J. Solid-State Circuits*, 46(1), 259–275.

[125] Posch, C., Serrano-Gotarredona, T., Linares-Barranco, B., & Delbruck, T. (2014). Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output. *Proc. IEEE*, 102(10), 1470–1484.

[126] Raffel, M. (2015). Background-oriented schlieren (BOS) techniques. *Exp. Fluids*, 56(3), 1–17.

[127] Raffel, M., Willert, C. E., Kompenhans, J., et al. (1998). *Particle image velocimetry: a practical guide*, volume 2. Springer.

[128] Rebecq, H., Gallego, G., Mueggler, E., & Scaramuzza, D. (2018). EMVS: Event-based multi-view stereo—3D reconstruction with an event camera in real-time. *Int. J. Comput. Vis.*, 126(12), 1394–1414.

[129] Rebecq, H., Horstschäfer, T., Gallego, G., & Scaramuzza, D. (2017). EVO: A geometric approach to event-based 6-DOF parallel tracking and mapping in real-time. *IEEE Robot. Autom. Lett.*, 2(2), 593–600.

[130] Rebecq, H., Ranftl, R., Koltun, V., & Scaramuzza, D. (2019a). Events-to-video: Bringing modern computer vision to event cameras. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)*.

[131] Rebecq, H., Ranftl, R., Koltun, V., & Scaramuzza, D. (2019b). High speed and high dynamic range video with an event camera. *IEEE Trans. Pattern Anal. Mach. Intell.*

[132] Reinbacher, C., Munda, G., & Pock, T. (2017). Real-time panoramic tracking for event cameras. In *IEEE Int. Conf. Comput. Photography (ICCP)* (pp. 1–9).

[133] Richard, H. & Raffel, M. (2001). Principle and applications of the background oriented schlieren (BOS) method. *Meas. Sci. Technol.*, 12(9), 1576.

[134] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (pp. 234–241).

[135] Rosinol Vidal, A., Rebecq, H., Horstschaefer, T., & Scaramuzza, D. (2018). Ultimate SLAM? combining events, images, and IMU for robust visual SLAM in HDR and high speed scenarios. *IEEE Robot. Autom. Lett.*, 3(2), 994–1001.

[136] Rudin, L. I., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1–4), 259–268.

[137] Rueckauer, B. & Delbruck, T. (2016). Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor. *Front. Neurosci.*, 10(176).

[138] Scheerlinck, C., Barnes, N., & Mahony, R. (2018). Continuous-time intensity estimation using event cameras. In *Asian Conf. Comput. Vis. (ACCV)*.

[139] Schmidt, B. E. & Woike, M. R. (2021). Wavelet-based optical flow analysis for background-oriented schlieren image processing. *AIAA Journal*, 59(8), 3209–3216.

[140] Seok, H. & Lim, J. (2020). Robust feature tracking in dvs event stream using Bezier mapping. In *IEEE Winter Conf. Appl. Comput. Vis. (WACV)* (pp. 1647–1656).

[141] Sethian, J. (1999). *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press.

[142] Settles, G. S. (2001). *Schlieren and shadowgraph techniques: visualizing phenomena in transparent media*. Springer Science & Business Media.

[143] Settles, G. S. & Hargather, M. J. (2017). A review of recent developments in schlieren and shadowgraph techniques. *Meas. Sci. Technol.*, 28(4), 042001.

[144] Settles, G. S. & Liberzon, A. (2022). Schlieren and BOS velocimetry of a round turbulent helium jet in air. *Optics and Lasers in Eng.*, 156, 107104.

[145] Shiba, S., Aoki, Y., & Gallego, G. (2022a). Event collapse in contrast maximization frameworks. *Sensors*, 22(14), 1–20.

[146] Shiba, S., Aoki, Y., & Gallego, G. (2022b). Secrets of event-based optical flow. In *Eur. Conf. Comput. Vis. (ECCV)* (pp. 628–645).

[147] Shiba, S., Aoki, Y., & Gallego, G. (2023a). Fast event-based optical flow estimation by triplet matching. *IEEE Signal Processing Letters*, 29, 2712–2716.

[148] Shiba, S., Aoki, Y., & Gallego, G. (2023b). A fast geometric regularizer to mitigate event collapse in the contrast maximization framework. *Advanced Intelligent Systems*, (pp. 2200251).

[149] Sironi, A., Brambilla, M., Bourdis, N., Lagorce, X., & Benosman, R. (2018). HATS: Histograms of averaged time surfaces for robust event-based object classification. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 1731–1740).

[150] Sommersel, O., Bjerketvedt, D., Christensen, S., Krest, O., & Vaagsaether, K. (2008). Application of background oriented schlieren for quantitative measurements of shock waves from explosions. *Shock Waves*, 18, 291–297.

[151] Son, B., Suh, Y., Kim, S., Jung, H., Kim, J.-S., Shin, C., Park, K., Lee, K., Park, J., Woo, J., Roh, Y., Lee, H., Wang, Y., Ovsiannikov, I., & Ryu, H. (2017). A 640x480 dynamic vision sensor with a 9μm pixel and 300Meps address-event representation. In *IEEE Intl. Solid-State Circuits Conf. (ISSCC)*.

[152] Stoffregen, T., Gallego, G., Drummond, T., Kleeman, L., & Scaramuzza, D. (2019). Event-based motion segmentation by motion compensation. In *Int. Conf. Comput. Vis. (ICCV)* (pp. 7243–7252).

[153] Stoffregen, T. & Kleeman, L. (2017). Simultaneous optical flow and segmentation (SOFAS) using Dynamic Vision Sensor. In *Australasian Conf. Robot. Autom. (ACRA)*.

[154] Stoffregen, T. & Kleeman, L. (2019). Event cameras, contrast maximization and reward functions: an analysis. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 12292–12300).

[155] Stoffregen, T., Scheerlinck, C., Scaramuzza, D., Drummond, T., Barnes, N., Kleeman, L., & Mahony, R. (2020). Reducing the sim-to-real gap for event cameras. In *Eur. Conf. Comput. Vis. (ECCV)* (pp. 534–549).

[156] Suh, Y., Choi, S., Ito, M., Kim, J., Lee, Y., Seo, J., Jung, H., Yeo, D.-H., Namgung, S., Bong, J., seok Kim, J., Park, P. K. J., Kim, J., Ryu, H., & Park, Y. (2020). A 1280x960 Dynamic Vision Sensor with a 4.95-μm pixel pitch and motion artifact minimization. In *IEEE Int. Symp. Circuits Syst. (ISCAS)*.

[157] Sun, D., Roth, S., & Black, M. J. (2013). A quantitative analysis of current practices in optical flow estimation and the principles behind them. *Int. J. Comput. Vis.*, 106(2), 115–137.

[158] Taverni, G., Moeys, D. P., Li, C., Cavaco, C., Motsnyi, V., Bello, D. S. S., & Delbruck, T. (2018). Front and back illuminated Dynamic and Active Pixel Vision Sensors comparison. *IEEE Trans. Circuits Syst. II*, 65(5), 677–681.

[159] Tedaldi, D., Gallego, G., Mueggler, E., & Scaramuzza, D. (2016). Feature detection and tracking with the dynamic and active-pixel vision sensor (DAVIS). In *Int. Conf. Event-Based Control, Comm. Signal Proc. (EBCCSP)*.

[160] Teed, Z. & Deng, J. (2020). RAFT: Recurrent all pairs field transforms for optical flow. In *Eur. Conf. Comput. Vis. (ECCV)* (pp. 402–419).

[161] Tian, Y. & Andrade-Cetto, J. (2022). Event transformer flownet for optical flow estimation. In *British Mach. Vis. Conf. (BMVC)*.

[162] Trucco, E. & Verri, A. (1998). *Introductory Techniques for 3-D Computer Vision*. Upper Saddle River, NJ, USA: Prentice Hall PTR.

[163] Tu, J. H., Rowley, C. W., Luchtenburg, D. M., Brunton, S. L., & Kutz, J. N. (2014). On dynamic mode decomposition: Theory and applications. *J. Computational Dynamics*, 1(2), 391–421.

[164] Tulyakov, S., Bochicchio, A., Gehrig, D., Georgoulis, S., Li, Y., & Scaramuzza, D. (2022). Time lens++: Event-based frame interpolation with parametric non-linear flow and multi-scale fusion. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 17755–17764).

[165] Venkatakrishnan, L. & Meier, G. (2004). Density measurements using the background oriented schlieren technique. *Experiments in Fluids*, 37, 237–247.

[166] Wang, X., Li, J., Zhu, L., Zhang, Z., Chen, Z., Li, X., Wang, Y., Tian, Y., & Wu, F. (2021). Visevent: Reliable object tracking via collaboration of frame and event flows. *IEEE Trans. Cybern.*

[167] Wang, Y., Idoughi, R., & Heidrich, W. (2020a). Stereo event-based particle tracking velocimetry for 3D fluid flow reconstruction. In *Eur. Conf. Comput. Vis. (ECCV)* (pp. 36–53).

[168] Wang, Z. W., Duan, P., Cossairt, O., Katsaggelos, A., Huang, T., & Shi, B. (2020b). Joint filtering of intensity images and neuromorphic events for high-resolution noise-robust imaging. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 1606–1616).

[169] Willert, C. E. & Klinner, J. (2022). Event-based imaging velocimetry: an assessment of event-based cameras for the measurement of fluid flows. *Exp. Fluids*, 63(6), 1–20.

[170] Ye, C., Mitrokhin, A., Parameshwara, C., Fermüller, C., Yorke, J. A., & Aloimonos, Y. (2020). Unsupervised learning of dense optical flow, depth and egomotion with event-based sensors. In *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)* (pp. 5831–5838).

[171] Zhang, J., Yang, X., Fu, Y., Wei, X., Yin, B., & Dong, B. (2021). Object tracking by jointly exploiting frame and event domain. In *Int. Conf. Comput. Vis. (ICCV)* (pp. 13043–13052).

[172] Zhang, X. & Yu, L. (2022). Unifying motion deblurring and frame interpolation with events. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 17765–17774).

[173] Zhang, Z., Yezzi, A., & Gallego, G. (2022). Formulating event-based image reconstruction as a linear inverse problem with deep regularization using optical flow. *IEEE Trans. Pattern Anal. Mach. Intell.*, (pp. 1–18).

[174] Zhou, Y., Gallego, G., Lu, X., Liu, S., & Shen, S. (2021a). Event-based motion segmentation with spatio-temporal graph cuts. *IEEE Trans. Neural Netw. Learn. Syst.*, (pp. 1–13).

[175] Zhou, Y., Gallego, G., & Shen, S. (2021b). Event-based stereo visual odometry. *IEEE Trans. Robot.*, 37(5), 1433–1450.

[176] Zhu, A. Z., Atanasov, N., & Daniilidis, K. (2017a). Event-based feature tracking with probabilistic data association. In *IEEE Int. Conf. Robot. Autom. (ICRA)* (pp. 4465–4470).

[177] Zhu, A. Z., Atanasov, N., & Daniilidis, K. (2017b). Event-based visual inertial odometry. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 5816–5824).

[178] Zhu, A. Z., Thakur, D., Ozaslan, T., Pfrommer, B., Kumar, V., & Daniilidis, K. (2018a). The multivehicle stereo event camera dataset: An event camera dataset for 3D perception. *IEEE Robot. Autom. Lett.*, 3(3), 2032–2039.

[179] Zhu, A. Z., Yuan, L., Chaney, K., & Daniilidis, K. (2018b). EV-FlowNet: Self-supervised optical flow estimation for event-based cameras. In *Robotics: Science and Systems (RSS)*.

[180] Zhu, A. Z., Yuan, L., Chaney, K., & Daniilidis, K. (2019). Unsupervised event-based learning of optical flow, depth, and egomotion. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 989–997).

[181] Zou, Y., Zheng, Y., Takatani, T., & Fu, Y. (2021). Learning to reconstruct high speed and high dynamic range videos from events. In *IEEE Conf. Comput. Vis. Pattern Recog. (CVPR)* (pp. 2024–2033).