Self-Supervised Learning of Event-Based Optical Flow via Deep Equilibrium Models

MSc Thesis Control & Operations

Aleksandar Shokolarov

TU Delft

Self-Supervised Learning of Event-Based Optical Flow via Deep Equilibrium Models

MSc Thesis Control & Operations

by

Aleksandar Shokolarov

to obtain the degree of Master of Science at the Delft University of Technology
to be defended publicly on Wednesday, May 29th, 2024 at 09:30

Thesis committee	
Chair:	Dr. ir. C. de Wagter
Supervisors:	Dr. G.C.H.E. de Croon
	Yilun Wu
External examiner:	Dr. ir. R. Sabzevari
Place:	Faculty of Aerospace Engineering, Delft
Project Duration:	May 2023 - May 2024
Student number:	4836839

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Faculty of Aerospace Engineering · Delft University of Technology

Preface

This thesis report represents the culmination of my research work over the past year. The journey towards it has been both challenging and rewarding, but none of it would have been possible without the following people.

Firstly, I would like to thank my supervisors, who guided me in this project. A big thank you to Yilun, who was an invaluable source of knowledge and provided me with directions whenever I felt lost. And for Guido, for the different perspectives and the warm encouragement you offered. I feel fortunate to have worked under the supervision of both of you, and I thank you for that.

I want to thank my closest friends, with whom I have shared countless moments of childish carelessness and happiness and who have provided me with endless optimism for the future. Your support is what kept me motivated during the last almost six years, and I hope this degree will always remind me of that. I deeply cherish and appreciate you.

Finally and most importantly, I would like to thank my family. Благодаря ви, че ми дадохте възможността да съм тук днес и цялата подкрепа, която сте ми предоставили по пътя. Без тях нямаше да мога и да си представя на какво съм способен и да стана човека, който съм. Надявам се някой ден да мога да ви върна и частица от всичко това. Обичам ви!

Aleksandar Shokolarov
Delft, May 2024

Abstract

The estimation of optical flow, which determines the movement of objects in a visual scene, is a crucial problem in computer vision. It is essential for applications such as autonomous navigation, where precise motion estimation is critical for performance and safety.

Frame-based cameras capture sequences of still images at regular intervals, from which optical flow is traditionally extracted using optimization-based or learning-based methods. Recently, event-based cameras, which detect changes in pixel brightness asynchronously, have gained traction due to their high temporal resolution and robustness to motion blur, and many algorithms have been developed to estimate optical flow from this data. IDNet is a learning-based approach that achieves state-of-the-art performance. However, IDNet and similar models face two major challenges: they require labeled ground-truth data for training, which is scarce and difficult to collect, and they rely on recurrent neural networks (RNNs) with a fixed number of refinement iterations. This fixed iteration scheme does not adapt to scene complexity, limiting accuracy for complex flows and increasing computational effort for simpler patterns.

The aim of this project is to explore, implement, and evaluate potential methods to address these two mentioned limitations and enhance the capabilities of models like IDNet.

To remove the need for ground-truth data, a self-supervised learning paradigm was implemented by introducing a novel contrast maximization loss that assesses the blur present when accumulating raw events for a certain time interval and compensating it with the predicted flow. To assess the effectiveness of this method, models were trained on the benchmark MVSEC dataset, showing improved results over previous methods with up to 15% on some sequences and an 8% improvement on average. Based on these experiments and results, further research directions were proposed.

As for the problem of the current fixed iteration scheme, Deep Equilibrium Models were found to provide a promising pathway to solving it. These novel models reformulate their iterative structure into a root-finding problem and utilize traditional solvers to find a solution based on some tolerance, providing a trade-off between speed and accuracy. Moreover, they allow for direct differentiation through the network using only their final estimate, compared to previous methods that keep track of their state through all iterations, leading to a *O(1)* memory consumption. Utilizing these and some additional ideas, the trained DEQ IDNet model reached competitive performance on DSEC while consuming 15% less memory. Yet, further work is needed to close the gap and achieve state-of-the-art performance.

Contents

Nomenclature

List of Abbreviations

AAE	Average Angular Error
AEE	Average Endpoint Error
AI	Artificial Intelligence
ANN	Artificial Neural Network
BPTT	Backpropagation Through Time
CM	Contrast Maximization
CNN	Convolutional Neural Network
DEQ	Deep Equilibrium
DNN	Deep Neural Network
GPS	Global Positioning System
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
ID	Iterative Deblurring
IFT	Implicity Function Theorem
IMU	Inertial Measurement Unit
IWE	Image of Warped Events
nPE	n-pixel Outlier Percentage
RNN	Recurrent Neural Network
SGD	Stochastic Gradient Descent
SLAM	Simultaneous Localization and Mapping
SOTA	State-Of-The-Art
SSL	Self-Supervised Learning
TMA	Temporal Motion Aggregation
UAV	Unmanned Aerial Vehicle
WMS	Warm-Starting Module

List of Symbols

Δ	Laplacian operator
ϵ	Small constant used for numerical stability
μ	Mean value
∇	Gradient operator
I	Image of warped events
I_{xx}	Laplacian in x-direction
I_x	Gradient in x-direction
I_{yy}	Laplacian in y-direction
I_y	Gradient in y-direction
J	Jacobian matrix
t_{begin}	Beginning of event accumulation interval
t_{end}	End of event accumulation interval
t_{ref}	Reference time
u	Optical flow component in the x-direction
v	Optical flow component in the y-direction

List of Figures

1 Introduction

1.1. Background

The estimation of optical flow is a fundamental problem in computer vision that deals with determining the movement of objects in a visual scene. This concept is vital in many applications such as object detection and segmentation [1] as well as autonomous navigation [2], where precise motion estimation is crucial for optimal performance and safety.

Frame-based cameras capture entire images at regular intervals, known as frames, creating a sequence of still images over time. Optical flow can then be extracted from these frames using optimization-based [3, 4] or learning-based methods [5, 6]. More recently, event-based cameras [7], which capture changes in pixel brightness asynchronously, have gained traction in the scientific community. This is due to their high temporal resolution, wide dynamic range, and motion blur robustness.

This data format is specifically well-posed for the problem of optical flow determination due to the continuous nature of the data. And many algorithms that tackle the estimation of optical flow from event-based cameras have already been developed [8, 9]. IDNet [10] is a learning-based model that relies on inductive priors to extract optical flow directly from the motion traces of the events, allowing it to achieve state-of-the-art performance. However, IDNet and other similar models suffer from two major drawbacks. Firstly, they rely on labeled ground-truth data during training, which is scarce and very hard to collect, preventing them from scaling and improving their accuracy further. Moreover, these models are centered around recurrent neural networks (RNN) that refine their prediction a fixed number of times. This fixed iteration scheme does not adapt to the complexity of the scene, which prevents it from estimating complex flow fields accurately and also from saving computational effort on simple flow patterns.

1.2. Research Formulation

The goal of this project is to research, implement, and assess possible methods for dealing with these two aforementioned limitations and expanding the capabilities of models such as IDNet. Therefore, the research objective can be briefly summarized as:

Research Objective

Expand the capabilities of IDNet by removing its dependency on labeled data and reforming its refinement scheme.

Since the two problems are orthogonal to each other, subdividing them allows for independent research in both directions. As such, the first research question that will be investigated is:

Research Question 1

Can a self-supervised training paradigm be applied to IDNet?

As for the direction of the second problem, which concerns the refinement scheme of IDNet, the research question can be formulated as follows:

Research Question 2

Can the fixed-iteration refinement structure employed by IDNet be replaced by an adaptive one?

1.3. Structure of the Report

This report will be separated into two main sections:

1. Part I will present the scientific paper that summarizes all the key findings of this research project.
2. Part II includes all the preliminary steps that were taken to define the direction of this project, which are further subdivided into the following. Chapter 3 presents the literature review that was carried out to investigate the principles of optical flow estimation and event-based cameras as well as self-supervised learning paradigms and adaptive inference mechanisms. Chapter 4 showcases the experiments that were carried out to investigate the different applications of self-supervised learning and the results obtained. Finally, Chapter 5 gives a conclusion to this report, and Chapter 6 summarizes the recommendations for further research.

Part I

Scientific Article

Self-Supervised Learning of Event-Based Optical Flow via Deep Equilibrium Models

Aleksandar Shokolarov
Faculty of Aerospace Engineering
Technical University of Delft
Delft, the Netherlands
Email: A.K.Shokolarov@student.tudelft.nl

Yilun Wu
Faculty of Aerospace Engineering
Technical University of Delft
Delft, the Netherlands
Email: Y.Wu-9@tudelft.nl

G.C.H.E. de Croon
Faculty of Aerospace Engineering
Technical University of Delft
Delft, the Netherlands
Email: G.C.H.E.deCroon@tudelft.nl

***Abstract*—Current state-of-the-art methods for event-based optical flow estimation rely on learning-based models trained on massive amounts of labeled data. However, the process of generating such data demands extensive labor and is prone to errors. Moreover, these methods are usually based on recurrent models that perform a fixed number of L iterations to refine their flow estimate, which has huge requirements in terms of computation and memory. As such, they also cannot adapt to the complexity of scenes, which may require fewer iterations to converge. We propose two improvements over these methods to deal with these aforementioned problems. Firstly, we introduce a self-supervised training framework that requires only event data during training and removes the need for costly ground-truth. Secondly, we adopt a deep equilibrium (DEQ) formulation that directly solves for the "equilibrium" flow, allowing models to adaptively change the number of iterations based on the flow complexity while decreasing the memory complexity from $\mathrm{O(L)}$ to $\mathrm{O(1)}$ during training. We also introduce a novel tolerance scheduling algorithm that allows our DEQ models to adapt their solver tolerance, allowing them to progressively converge to better solutions. We train and test our methods on the DSEC and MVSEC datasets and compare them to current state-of-the-art models, showcasing better estimation accuracy by 8% on average for our self-supervised model and a 15% reduction in memory consumption for our DEQ model while providing competative performance.**

1. Introduction

Optical flow is a fundamental concept in computer vision that refers to the pattern of apparent motion of pixels, objects, surfaces, and edges in a visual scene as recorded by a camera or other optical sensor. Formally, it assigns a velocity vector, described by a direction and a magnitude, for each pixel in an image. In simpler terms, it tries to describe the movement of objects in an image from one frame to the next.

As such, optical flow plays a critical role in many real-world applications ranging from object detection and segmentation [1] to the navigation of aerial robots [2]. However, achieving high accuracy requires sophisticated algorithms

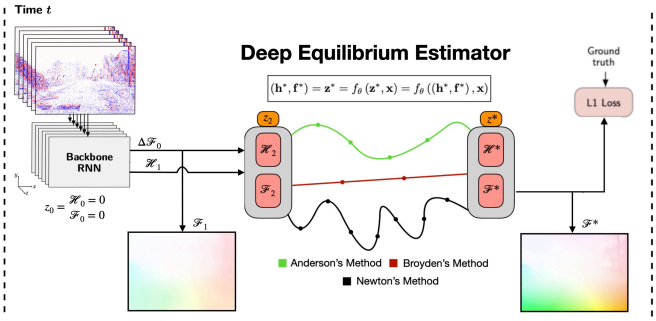


Figure 1: A visual representation of the IDNet DEQ architecture, which computes the "equilibrium" flow directly from the raw events by utilizing a black-box solver such Anderson's or Newton's method.

that can handle real-world imperfections such as brightness changes across scenes and sensor noise. When extracted from frame-based cameras, optical flow has traditionally been estimated using optimization-based methods [3], [4] algorithms, while more recently, with the advancement of deep neural networks, learning-based methods [5], [6], [7], [8] have also been successfully applied to the problem. However, these methods still struggle to deal with the inherent problems of frame-based cameras, such as motion blur and over-saturation.

Instead of capturing frames at a specific framerate, event-based cameras detect changes in luminance asynchronously at the pixel level, generating "events" only when a significant change occurs [9]. Consequently, event-based cameras boast exceptional temporal resolution, operating within the microsecond range, granting improved robustness to motion blur. Additionally, their remarkable dynamic range enables them to capture scenes characterized by substantial variations in brightness and illumination.

Due to these qualities, event-based cameras hold an enormous potential to revolutionize the field of computer vision, especially related to the problem of determining optical flow since the trajectory of pixels is continuously encoded in the data instead of discretely between two frames. However, their novel data format renders traditional estimation

methods that operate on frames obsolete. Therefore, there has been a significant push to develop new algorithms that exploit the structure and inherent benefits of event-based data to estimate optical flow.

Learning-based methods that utilize convolutional networks [10] have cemented themselves as the standard approach for estimating optical flow from event-based data. Building on that, recent works [11] make use of recurrent structures to enhance their estimations. These models commonly utilize a Gated Recurrent Unit (GRU) to improve their optical flow estimate iteratively, drawing inspiration from conventional optimization-based techniques [3], [4].

One such example is IDNet [12], which builds on the observation that by accumulating events onto an image, the motion of objects can be directly extracted from the blur present in that image, which can better understood visually through Figure 2. By integrating the idea of motion compensation [13], IDNet also refines its estimate of the optical flow by iteratively "deblurring" the event representations and processing them using a recurrent unit, allowing it to achieve state-of-the-art performance on the both DSEC [14] and MVSEC [15].

However, the current training paradigm still requires vast amounts of labeled data to achieve such performance. In reality, collecting ground truth optical flow is a process that requires immense effort and time, especially in the presence of occlusions, motion discontinuities, or non-rigid deformations. These factors can lead to ambiguities in flow estimation and make it difficult to obtain accurate ground truth. This reliance on labeled data currently stands as a bottleneck because it not only hinders the training of bigger models but also limits the generalizability of existing ones. Moreover, in its current configuration, IDNet uses a predefined fixed amount of L iterations to refine its optical flow estimate, regardless of the complexity of the scene and the amount of motion present. This also involves training via the backpropagation-through-time (BPTT) algorithm [16], which needs to keep track of all L iterations during training. This brings a significant computational burden, setting a limit on the model size and refinement iterations. Ideally, having a mechanism that adaptively changes the number of iterations required to converge to a flow estimate could lead to reduced latency, which is critical when considering real-time applications, but could also improve the accuracy on scenes with complex flow fields.

In this work, we propose two advancements over the original IDNet architecture, which aim to push its performance to new heights. Firstly, removing its dependency on ground truth optical flow by converting to a self-supervised training paradigm that utilizes a contrast maximization framework [17], [18] at its core and allowing the network to learn directly from the raw event data. Secondly, the fixed-step recurrent update architecture is replaced by a Deep Equilibrium (DEQ) estimator that integrates recent progress in implicit deep learning [19], [20], [21]. This step brings two substantial benefits to IDNet. Namely, it allows the network to adaptively change the number of iteration steps needed to converge to the final flow estimate, improving both speed and accuracy. On top of that, it removes the need for the costly BPTT, which not only significantly reduces the memory requirements during training, but also significantly speeds up the computation of the backward pass.

Following this section, in section 2 the literature investigation that was carried out to build a theoretical base for this research will be presented. section 3 will describe the methods that were developed to achieve our research objective. Next in section 4, information will be given about the experiments that were conducted as well as the results obtained from them. Finally, section 5 will conclude this article, summarizing the most important findings and giving recommendations for future work.

2. Related Work

Optical Flow Estimation

The estimation of optical flow stands as a cornerstone problem in computer vision, spanning multiple decades of investigation. In the early 1980s, the pioneering work of Horn and Schunck [4] laid the groundwork for what has been the standard algorithm for determining optical flow. Their seminal paper introduced a method that computed optical flow by minimizing an energy function based on two fundamental assumptions:

1) **Brightness constancy:** the brightness intensity of pixels remains constant over time.
2) **Spatial smoothness:** neighboring pixels tend to have similar motion patterns.

which are used to formulate the terms in the energy function. This energy function is then minimized through iterative procedures such as gradient descent to produce a dense optical flow field, representing the motion vectors for each pixel in the image. While the Horn-Schunck method was groundbreaking, it had limitations, particularly in handling large displacements, occlusions, and complex motion patterns. Therefore, in the following years, researchers explored various approaches to address these limitations, such as using pyramidal structures [22], enabling them to handle different levels of detail and motion intricacies. Still, the added complexity significantly increased the latency of computing flow using that approach, limiting its use in real-time applications.

More recently, learning methods [5], [6] have taken over and replaced traditional methods partly by mimicking their operation. PWC-Net [7] borrows the ideas of pyramidal structures, warping, and cost volumes to outperform its predecessors at a fraction of the network size. RAFT [8] introduces a new paradigm of employing a recurrent neural network (RNN). At its core, it builds cost volumes containing correlations between all pairs of pixels, which are then fed through the recurrent unit to update the flow estimate incrementally. By iteratively refining flow estimation through recurrent updates, it encourages the model to generate flow fields that are consistent across time steps. This leads to

smoother and more temporally coherent flow predictions, allowing the architecture to outperform its predecessors substantially.

Due to the promising advantages of event-based cameras and the advent of publicly available datasets [14], [15], learning-based methods have also been used to tackle the problem of estimating optical flow from event data. [17] propose an encoder-decoder architecture that utilizes a novel event representation, which temporally discretizes events into bins and processes them to arrive at a flow estimate. E-RAFT [11] employs the same strategy as RAFT [8] to construct correlation volumes, treating the event representations as frames. These representations are again fed through an RNN to refine the final prediction progressively. [23] builds on these ideas and introduces an additional attention module to emphasize motion patterns that are consistent with each other, outperforming E-RAFT. However, the computation and storage of the correlation volumes used by both models greatly increase their memory consumption and inference time, diminishing the possibility of deploying the architectures on real-time platforms. IDNet [12] targets this problem by directly eliminating the correlation volumes and instead observing that the motion of objects is directly encoded in the blur present in the raw events. Moreover, IDNet iteratively refines its optical flow estimate by "deblurring" the event representations and processing them using a recurrent unit, reaching state-of-the-art performance.

Self-Supervised Learning

Despite the success of these methods in utilizing the novel event data format to obtain accurate predictions of optical flow, most of them still require substantial amounts of annotated ground truth data to train their underlying neural networks. Unfortunately, creating such datasets is extremely laborious, subject to errors, and time-consuming; therefore, eliminating it is greatly beneficial to creating even bigger datasets that would allow for the training of more complex models.

EV-FlowNet [10] formulates the training regime as a self-supervised problem using only event data and a set of corresponding grayscale images, corresponding to the beginning and end of the optical flow interval. It then warps the second image back to the first one with the idea that as the optical flow prediction converges to the true optical flow, the warped and the original images will become identical. The photometric loss between the warped and original images is computed to measure their dissimilarity, and a smoothness term is added. The method, while successful, still requires a set of corresponding grayscale images on top of the event data to generate the supervisory signal, which is suboptimal since the latency of the frame-based camera is usually much higher than that of the event-based camera. Zhu et al. [17] remove the dependency on images by minimizing the per-pixel, per-polarity average timestamp of the warped events. This is also known as contrast maximization since the better the flow estimate is, the higher the alignment or "contrast" of the warped events is. Hagenaars et al. [18] improve on that by scaling the loss by the sum of pixels with at least one warped event to convexify the metric and improve training performance.

Deep Equilibrium Models

Traditional optical flow estimation methods [4] rely on an energy minimization principle that is optimized until a stable flow field is produced. Recurrent networks such as IDNet mimic this principle by iteratively refining their estimate. A drawback of this method is that the iteration is carried out for a fixed number of steps, regardless of the flow quality. But what if this iterative procedure could be replaced with an architecture that can adaptively converge to a stable solution, and without the need for BPTT?

This is exactly the idea behind Deep Equilibrium [19] models. By re-formulating the architecture as an infinitely deep implicit layer, the outputs of the model become the fixed points of this new implicit layer, which can be solved by using any traditional root-finding method. The quality of the converged solution depends on the tolerance used by the solver, which allows the network to adaptively change the number of iterations needed, closely resembling traditional optimization-based methods. Moreover, by making use of the Implicit Function Theorem (IFT) [19], differentiating such a network requires no knowledge about the intermediate iterations but only about the final fixed point, allowing for constant memory training, regardless of the number of iterations performed.

Bai et all. [20] apply this implicit layer approach to the RAFT [8] architecture, achieving faster flow convergence and improved accuracy while using 4-6 times less training memory. The authors also explore the questions of DEQ stability and training efficiency, successfully integrating previous [21] and new ideas to address them.

3. Methodology

Preliminaries

The problem of optical flow estimation is characterized by mapping a sequence of events within an interval $[t_{begin}, t_{end}]$ to an optical flow vector (u, v), belonging to every pixel. Each event comprises a timestamp t, denoting when it was triggered, pixel coordinates x and y, and binary polarity p. IDNet [12] utilizes the event representation introduced by [17], which temporally discretizes a set of events $e_i = (t_i \; x_i \; y_i \; p_i)$ into B bins by linearly interpolating each event into its two adjacent bins weighted by polarity and timestamp. This allows to create the event representation $\mathcal{E} \in R^{B \times H \times W}$ using:

$$t_i^* = (B-1)\,(t_i - t_{begin}) \,/\, (t_{begin} - t_{end}) \tag{1}$$

$$\mathcal{E}(B,x,y) = \sum_{i|x_i=x,y_i=y} p_i \max\left(0, 1 - |B - t_i^*|\right) \tag{2}$$

The results of this interpolation process generate a voxel grid tensor, allowing each bin to be treated as a separate 2D image.

Self-Supervised Learning

To convert the supervised training scheme used by IDNet to a self-supervised one that requires no ground truth, we utilize the contrast maximization framework described by [17], [18], with the idea of learning the optical flow directly from the event stream. This optimization framework operates on the principle that optical flow information is contained within the spatial and temporal discrepancies among events generated by a moving edge, or more obviously - the blur in the accumulated raw events. To access this information, we apply motion compensation to the raw events [13], by transporting each event using the predicted optical flow. Having an estimate of the flow $\mathbf{u}(\mathbf{x}) = (u, v)^T$, the events with coordinates $\mathbf{x}_i = (x_i, y_i)$ can be propagated to a reference time t_{ref} using:

$$\begin{pmatrix} x_i' \\ y_i' \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \end{pmatrix} + (t_{ref} - t_i) \begin{pmatrix} u(x_i, y_i) \\ v(x_i, y_i) \end{pmatrix} \tag{3}$$

Under perfect optical flow estimation, transporting these events would have the effect of eliminating the motion blur as presented in Figure 2.

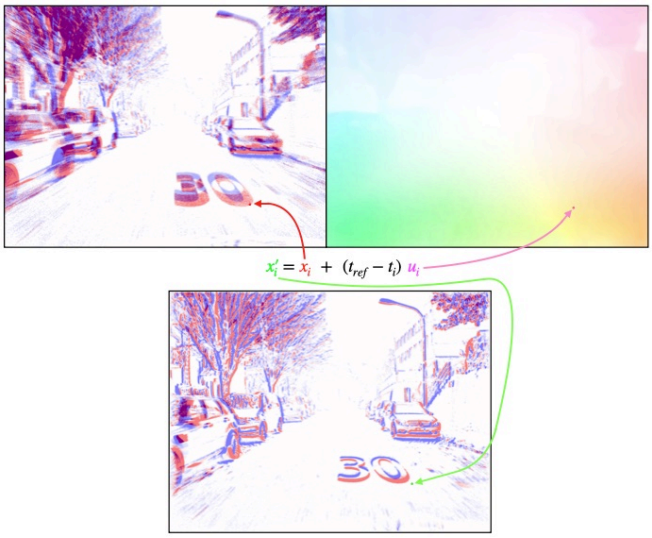


Figure 2: Showcasing the effect of motion compensating a set of accumulated raw events(left) using a good flow estimate(right), generating a set of events with minimal blur present(bottom).

To measure the quality of this deblurring process, [18] propose calculating the per-pixel and per-polarity average timestamp of the image of warped events (IWE), which is generated by motion-compensating the raw events and accumulating them onto an image. The lower the value of this metric, the better the flow predictions are. To compute it, we create an image where each pixel represents the average timestamp for each polarity T_+, T_-.

$$T_{p'}(\boldsymbol{x}; \boldsymbol{u} \mid t_{\text{ref}}) = \frac{\sum_j \kappa\left(x - x_j'\right) \kappa\left(y - y_j'\right) t_j}{\sum_j \kappa\left(x - x_j'\right) \kappa\left(y - y_j'\right) + \epsilon} \tag{4}$$

$$\kappa(a) = \max(0, 1 - |a|) \tag{5}$$

During the training process, [17] use the sum of the two images squared as a loss metric. However, [18] note that it exhibits non-convex behavior, which is undesirable, and therefore, they scale it by the sum of pixels with at least one warped event to convexify it, resulting in:

$$\mathcal{L}_{\text{contrast}}\left(t_{\text{ref}}\right) = \frac{\sum_{\boldsymbol{x}} T_+(\boldsymbol{x}; \boldsymbol{u} \mid t_{\text{ref}})^2 + T_-(\boldsymbol{x}; \boldsymbol{u} \mid t_{\text{ref}})^2}{\sum_{\boldsymbol{x}} \left[n\left(\boldsymbol{x}'\right) > 0\right] + \epsilon} \tag{6}$$

where $n(\mathbf{x}')$ counts the per-pixel events in the IWE. Another problem that arises with this formulation is that transporting to a single reference time introduces a scaling challenge, as the output flows u and v are influenced by the scale factor $(t_{ref} - t_i)$. Consequently, during backpropagation, gradients are weighted more heavily towards events with timestamps farther from t_{ref}, while events occurring very close to it are effectively disregarded. To fix that, we warp the events to t_0 and t_{N-1}, allowing all events to contribute equally.

$$\mathcal{L}_{\text{contrast}} = \mathcal{L}_{\text{contrast}}(t_0) + \mathcal{L}_{\text{contrast}}(t_{N-1}) \tag{7}$$

Since optimizing this term directly could lead to non-physical solutions, a regularizing term is usually used to enforce this physicality. The Charbonnier smoothness term aims to do exactly so by forcing the assumption that neighboring pixels should have similar flow magnitudes and directions. It is computed using:

$$\ell_{\text{smoothness}} = \sum_{x,y} \sum_{i,j \in \mathcal{N}(x,y)} \rho(u(x,y) - u(i,j)) + \rho(v(x,y) - v(i,j)) \tag{8}$$

$$\rho(x) = \sqrt{x^2 + \epsilon^2} \tag{9}$$

where ϵ is a small constant added to prevent division by zero and improve numerical stability. Finally, the total loss used for optimization is:

$$\mathcal{L}_{\text{flow}} = \mathcal{L}_{\text{contrast}} + \lambda \mathcal{L}_{\text{smoothness}} \tag{10}$$

where λ is a coefficient used to weigh the smoothness term.

Deep Equilibrium Flow

The next step in this research was to convert the fixed-iteration scheme of IDNet to an adaptive DEQ formulation. For more details about its original architecture, refer to [12]. Looking at the internal workings of IDNet, we start by initializing the RNN at its core ($\mathbf{h_0}$) as well as the flow

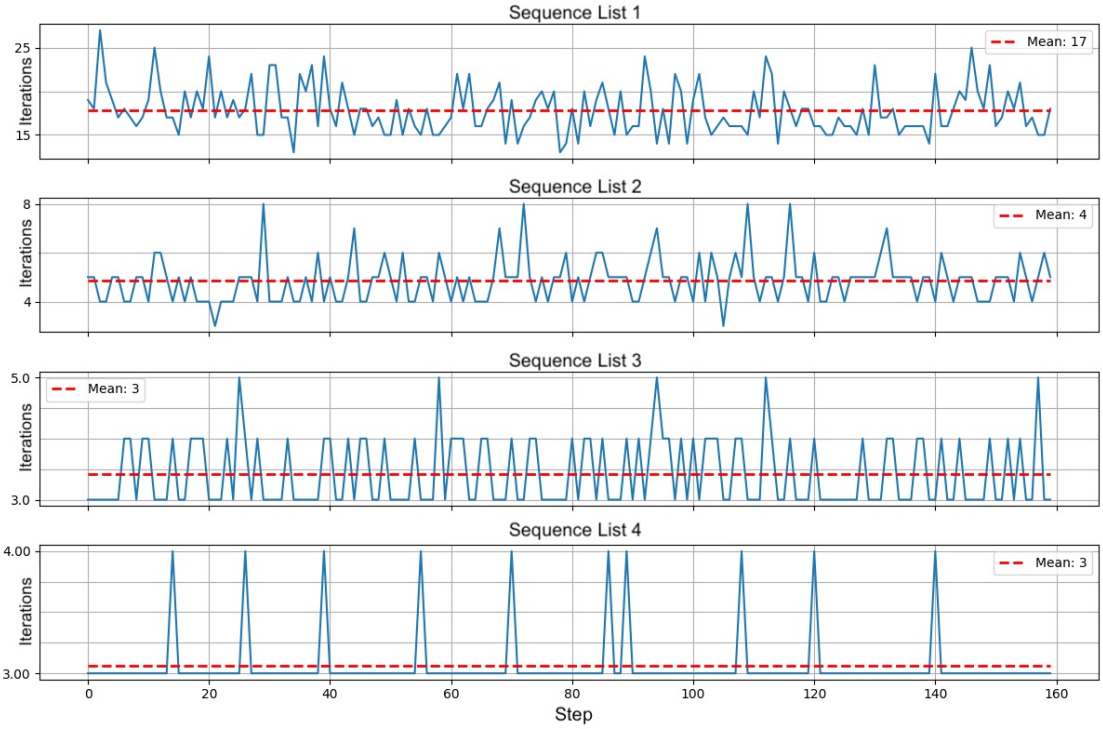


Figure 3: Figure showcasing the reduction of iterations needed during training for convergence on each successive sample from a sequence.

estimate ($\mathbf{f_0}$), setting their values to zero. What follows is an iteration process where the voxel bins containing the event information are deblurred and processed by the RNN, which is then used to refine the final flow. Additionally, we use a warm-starting module (WSM) to update the memory of the RNN. This defines a dynamical system of two states: 1) the state of the RNN $\mathbf{h}$, which is used to update the flow as the bins are processed, and 2) the final flow estimate $\mathbf{f}$. Formally, the dynamics of this system are described by:

$$\mathcal{E}_{deblur,i+1} = Deblur(\mathcal{E}_{deblur,i}; \mathbf{f_i}) \tag{11}$$

$$\mathbf{f}_{i+1} = \mathbf{f}_i + RNN(Encoder(\mathcal{E}_{deblur,i+1}); \mathbf{h}_i) \tag{12}$$

$$\mathbf{h}_{i+1} = WSM(\mathbf{f}_{i+1}) \tag{13}$$

This dynamical system can also be looked at as a fixed point equation whose solution represents the equilibrium point between the memory and flow states, characterized by:

$$(\mathbf{h}^*, \mathbf{f}^*) = \mathbf{z}^* = f_\theta(\mathbf{z}^*, \mathbf{x}) = f_\theta((\mathbf{h}^*, \mathbf{f}^*), \mathbf{x}) \tag{14}$$

where f_θ are the model parameters. $\mathbf{z}^*$ is the equilibrium solution, which is defined as the state that would incur no change even if we perform more steps of the fixed point equation. One of the most important advantages of this formulation is that compared to the traditional IDNet formulation, which naively takes steps through this operator, in this case, we can leverage the power of root-finding solvers, which find the state for which the change in z is zero, or in practice, below a certain tolerance. Advanced methods such as Broyden's method [24] or Anderson mixing [25] also can guarantee much faster convergence and better stability. Having obtained the equilibrium flow, we apply supervised learning by computing the loss with respect to the ground truth flow using an L1 loss:

$$\mathcal{L} = ||\mathbf{f}_{gt} - \mathbf{f}^*||_1 \tag{15}$$

Another big benefit of this formulation is that it allows us to differentiate this network using only the final equilibrium solution, disregarding any information about the trajectory, using the Implicit Function Theorem (IFT) [19].

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}^*} \left(I - \frac{\partial f_\theta}{\partial \mathbf{z}^*} \right)^{-1} \frac{\partial f_\theta(\mathbf{z}^*, \mathbf{x})}{\partial \theta} \tag{16}$$

Compared to the traditional method of training recurrent networks via backpropagation-through-time (BPTT), which requires $O(L)$ memory where L is the number of refinement iterations, the DEQ formulation requires $O(1)$ memory, leading to a theoretical memory reduction of a factor of L.

Apart from that, the tools of implicit deep learning allow us to improve the training procedure further by approximating the gradient update. Despite the memory reduction

brought by the IFT, inverting the Jacobian term in Equation 16 can become too costly to compute as the input and network sizes grow. Therefore, we follow recent efforts [20], [21], [26] that have aimed at reducing that computational burden even further by using an approximation of that inverted Jacobian term, replacing it with the identity matrix I as in Equation 17. In practice, we note that this grants an almost free backward pass compared to the forward pass.

$$\frac{\partial \mathcal{L}}{\partial \theta} \approx \frac{\partial \hat{\mathcal{L}}}{\partial \theta} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}^*} \frac{\partial f_\theta\left(\mathbf{z}^*, \mathbf{x}\right)}{\partial \theta} \tag{17}$$

Still, stability and convergence of DEQ models have traditionally been an open question. [19], [26], [27], [28] note that DEQ models tend to experience an oscillatory behavior, which prohibits proper convergence. This is problematic for models that require significant time to perform even a single pass through the network, especially during training, since initially, the flow predictions are of low quality, and thus, a great number of iterations are needed to reach convergence. Therefore, we adopt the fixed-point correction scheme introduced by [20], which aims to stabilize the convergence by sampling the trajectory of the solver, which includes all intermediate solutions, and computing a loss term with respect to each of the solver states. The motivation behind this idea is that converging to the true solution earlier would result in a smaller final loss, thus stimulating faster and more stable convergence. Starting with a convergence trajectory ($\mathbf{f}^{[0]}$, ..., $\mathbf{f}^{[i]}$, ..., $\mathbf{f}^*$) where $\mathbf{f}^{[0]}$ and $\mathbf{f}^*$ are the initial and final flow estimates, we take N evenly-distributed samples from that trajectory and compute a fixed-point correction term as per:

$$\mathcal{L}_{cor} = \gamma ||\mathbf{f}^{[i]} - \mathbf{f}_{gt}||_1 \tag{18}$$

with γ being a hyperparameter. We note that this addition greatly increases the training stability while adding no significant computational cost.

And yet performing ten or twenty forward passes through deep models such as IDNet [12] would still pose a time-infeasibility problem. To alleviate that, we looked at strategies to warm-start the solver, drawing inspiration from traditional optimization approaches [4] and current efforts [20]. Since the time interval for which events are accumulated is very small, we note that there is a high degree of similarity between adjacent event representations and also between the ground truth flow corresponding to these intervals. Therefore, using the prediction made at one timestep as an initial guess for the following one allows us to essentially recycle its information to warm-start our DEQ model. Formally, considering the fixed-point solution $\mathbf{z}_\mathbf{i}^*$ at timestamp t_i, we apply it as an initial guess $\mathbf{z}_{\mathbf{i+1}}^{[\mathbf{0}]}$ at t_{i+1}. This allows us to compute the full trajectory only for the flow at t_0 and subsequently warm-start the solver at every following flow prediction, which dramatically reduces the number of iterations needed. To allow for warm-starting during training while still creating batches with random samples, we formed sequence lists that contained batches with samples that were shifted one time-interval ahead. For example, sequence list 1 contained a batch with sample indices 1, 10, and 20. The predictions from this sequence were then used to warm-start the solver for sequence list 2, which contained a batch with sample indices 2, 11, and 21, and so on. The results from this process are shown in Figure 3. As seen there, the average number of iterations for samples with indices i, $i+1$, $i+2$, and $i+3$ reduced from 17 to 4, 3, and 3, respectively, showcasing the usefulness of this addition.

Another problem that we came across experimentally is that during training, initially, the solver would need to perform a high number of iterations in order to reach convergence. But as the network got better and better at estimating the flow, that number would drastically decrease and converge to 1-2 iterations at the end of the training process since the solver would reach its fixed tolerance almost instantly. While this behavior is expected, pushing the solver to reach lower and lower tolerances would allow it to produce better equilibrium points, improving the final accuracy. Therefore, we applied a scheduling procedure that would decrease the solver tolerance as the network underwent training. For example, utilizing Anderson mixing [25] with an initial $\epsilon = 1 \times 10^{-2}$, we computed a moving average of the number of iterations needed to pass that tolerance, and if that number went below a set threshold of K iterations, we reduced the tolerance by a factor of two, and this process would continue until the end of training, allowing us to improve the final performance of the network substantially.

4. Experiments & Results

Self-Supervised Learning

On the MVSEC [15] dataset, we train the self-supervised version of IDNet on the $outdoor\_day2$ sequence, providing about 11 minutes of event data captured while driving on public roads. To do so, we take partitions of 30000 events, which are converted to the voxel grid defined in Equation 2 with $B = 9$, and center cropped to a resolution of 256 $\times$ 256 pixels. To weigh the smoothness term in Equation 10, λ is given a value of 0.1. During optimization, we used Adam [29] paired with a onecycle learning scheduler [30] with a maximum learning rate of 4×10^{-4} and a batch size of 10. We trained for 50 epochs and implemented the entire training procedure in PyTorch [31].

We test the trained model on both the $indoor\_flying$ and $outdoor\_flying$ sequences and generate optical flow estimations that are scaled to the duration of 1 ($dt = 1$) and 4 ($dt = 4$) grayscale frames. This scaling is done using the following:

$$(u', v') = (u, v) \times dt/(t_N - t_0) \tag{19}$$

We provide metrics, including the average endpoint error (AEE) and the proportion of points where the AEE exceeds 3 pixels, specifically focusing on pixels with valid ground truth flow and at least one event. We compare against

	outdoor day1		indoor flying1		indoor flying2		indoor flying3	
dt = 1 frame	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier
Ours	0.51	0.1	**0.55**	**0.0**	**0.87**	**1.67**	**0.82**	**1.15**
EV-FlowNet-CM	**0.32**	**0.0**	0.58	**0.0**	1.02	4.0	0.87	3.0
EV-FlowNet-PL	0.49	0.2	1.03	2.2	1.72	15.1	1.53	11.9

	outdoor day1		indoor flying1		indoor flying2		indoor flying3	
dt=4 frames	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier	AEE	% Outlier
Ours	1.35	10.3	**2.17**	**21.2**	**3.58**	**39.9**	**3.15**	**35.2**
EV-FlowNet-CM	1.30	9.7	2.18	24.2	3.85	46.8	3.18	47.8
EV-FlowNet-PL	**1.23**	**7.3**	2.25	24.7	4.05	45.3	3.45	39.7

TABLE 1: Quantitative evaluation of our optical flow network compared to EV-FlowNet and UnFlow. For each sequence, Average Endpoint Error (AEE) is computed in pixels, % Outlier is computed as the percent of points with AEE < 3 pixels. dt=1 is computed with a time window between two successive grayscale frames, and dt=4 is between four grayscale frames.

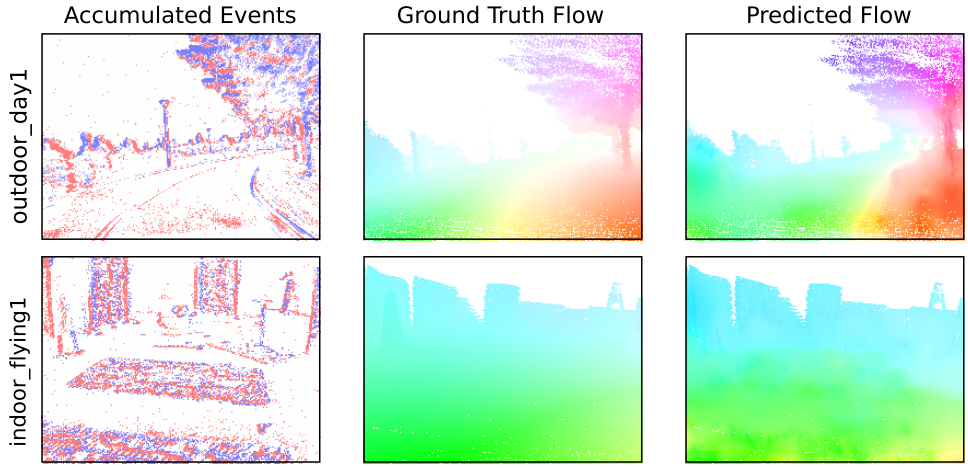


Figure 4: Comparison between ground truth optical flow(middle) and predicted optical flow(right) for one outdoor and one indoor sequence from MVSEC, with raw events shown(left).

two self-supervised versions of the EV-FlowNet model, EV-FlowNet-CM [17], which was training using the same contrast maximization framework as us, and EV-FlowNet-PL [10], which was trained using a photometric loss computed between the motion-compensated grayscale images. These metrics can be seen below in Table 1.

Quantitatively, we note that our model outperforms both EV-FlowNet-CM and EV-FlowNet-PL on all sequences except $outdoor\_day1$ in terms of AEE and outliers for both dt=1 and dt=4 frames. One problem that we came across by inspecting the results qualitatively is that the model has difficulty dealing with light sources such as street lamps, which would explain the slight under-performance on outdoor sequences. Yet, this showcases that our model can generalize to sequences with various motion complexities and magnitudes, providing state-of-the-art performance.

As for the qualitative analysis, we present a sample from an indoor and an outdoor sequence as shown in Figure 4. We see that, overall, the model is able to capture the magnitude and direction of the optical flow vectors well when compared to the ground truth. As for the finer details, the smooth transition of the flow from left to right is not as preserved, especially when looking at the outdoor sequence. Nevertheless, a lack of fine details is expected due to the self-supervised nature of the training process. Additional visualizations can be found in Appendix B.

Deep Equilibrium Flow

For the DEQ formulation of IDNet, we train on the DSEC [14] training set, consisting of 7800 samples captured while driving in day and night conditions. This time we adopt an event representation comprising 15 bins for every 100 milliseconds of events, synchronized with the available optical flow ground truth. We make use of the same optimization tools as previously; Adam [29] optimizer with onecycle learning schedule with a maximum learning rate of 1×10^{-4} and batch size of 4, and we train for 150 epochs. Additionally, we apply the fixed-point correction scheme by taking $N = 5$ samples evenly-spaced samples from the convergence trajectory, computing their loss with respect to the ground truth, and weighing them with $\gamma = 0.8$.

We test the trained model on the DSEC test set, which consists of 2100 samples. This time, we introduce the following additional metrics: AAE (Average Angular Error), representing the mean angular error of the optical flow

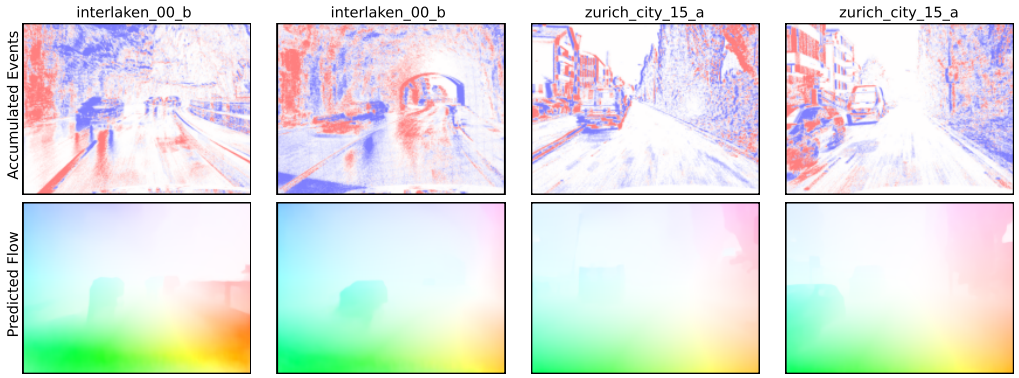


Figure 5: Qualitative results from evaluating the trained DEQ IDNet on the DSEC test set, showcasing snippets from two different environments - a mountain road environment and an urban one.

vector in degrees, and nPE (n-pixel Outlier Percentage), indicating the proportion of pixels with optical flow magnitude error exceeding n pixels. We compare the results to non-DEQ versions of IDNet [12]: IDNet(ID) and IDNet(TID), as well as MultiCM [32] and E-RAFT [11], summarizing the results in Table 2.

	AEE	AAE	1PE	2PE	3PE
MultiCM [32]	3.47	13.98	76.6	48.4	30.9
EV-FlowNet [10]	2.32	7.90	55.4	29.8	18.6
E-RAFT [11]	0.79	2.85	12.7	4.7	2.7
IDNet(ID) [12]	**0.72**	**2.72**	**10.1**	**3.5**	**2.0**
IDNet(TID) [12]	0.84	3.41	14.7	5.0	2.8
IDNet(DEQ)	0.92	3.68	16.1	5.5	3.1

TABLE 2: Quantitative evaluation of our DEQ network on the DSEC [14] test set and comparison to the results of similar models.

Quantitatively, the DEQ formulation of IDNet performs competitively, placing fourth behind the ID and TID versions of IDNet as well as E-Raft. Theoretically, a DEQ network can directly compute the gradient during a backward pass via the Implicit Function Theorem [19], regardless of the trajectory of the solver. This means that a DEQ version of IDNet should perform at least as well as a traditional one as long as the number of iterations performed is equal. Unfortunately, in practice, we find that this is not the case, which can be partially attributed to the approximation of the gradient as per [20], [21], [26]. This approximation degrades the quality of the training, leading to worse convergence and accuracy.

Qualitatively, we present four predictions taken out of two sequences from the test set as seen in Figure 5 that present two different scenarios: driving through mountain roads as shown in $interlaken\_00\_b$ and driving in an urban environment as shown in $zurich\_city\_15\_b$. In the first sequence, apart from smoothly capturing flow transitions from the left to the right of the scene, our model is able to restore sharp details, as evident in the car and road guard rail captured. As for the second sequence, the model is successfully able to pick up details in the foreground as well as the background, as seen in the silhouettes of the car and the minivan behind.

Ablation Study

In this section, we look more closely at the different ideas that were integrated to change the performance of the DEQ IDNet.

Warm-Starting. The warm-starting procedure was integrated to speed-up the convergence of the fixed-point solver and two strategies were considered. The first strategy, as mentioned in section 3, was to reuse the fixed-point solution from interval t_{i-1} as an initial guess at t_i, due to the high degree of correlation. This proved to work well, reducing the number of iterations needed for convergence with an order of magnitude. Another strategy, which was borrowed from the TID version of IDNet [12], was to directly predict the flow at the next time interval using a separate prediction head. Unfortunately, during training this proved to be infeasible since initially the prediction head produced results that were far off and were not suitable as an initial guess for the solver, again requiring a large number of iterations to reach convergence. Therefore, the first strategy was preferred.

Solver Tolerance. As for tolerance setting of the solver, we noticed a diminishing return as it was decreased, specifically below values of 1×10^{-3}. The solver tolerance is a measure of the relative difference between subsequent solutions, which represent the hidden state of the RNN - **h**, and the predicted optical flow - **f**. In practice, we found that this difference is mainly influenced by the predicted flow while the difference i hidden state of the RNN reached much lower values. This indicated that the relative differences below the aforementioned threshold value did not have a big effect on the fixed-point flow.

5. Conclusion

Recent works such as IDNet [12] show that supervised deep neural networks that utilize finite-step recurrent updates to refine their optical flow estimates are the current state-of-the-art in event-based optical flow estimation. In this work we present two improvements over the original IDNet architecture. Firstly, its supervised training procedure that requires vast amounts of ground truth data is replaced by a self-supervised framework that utilizes the idea of contrast maximization [17], [18] at its core. That allows IDNet to break free from its dependency on labeled data and train using only the event stream, allowing it to be potentially trained on more and more data. Moreover, we reformulate its finite-step recurrent unit using a deep equilibrium (DEQ) model that adaptively changes the number of iterations needed to arrive at a stable "equilibrium flow" which brings additional benefits in terms of memory consumption. We train and test the self-supervised version of IDNet on MVSEC [15], showcasing state-of-the-art performance compared to similar methods in self-supervised optical flow estimation. As for the DEQ reformulation, we train and test on DSEC [14] and note improvements in memory consumption by 15% while retaining competitive performance. We note that the gradient approximation algorithm results in non-optimal convergence, preventing us from reaching SOTA performance, and is, therefore, an area where more research is necessary. This indicates a promising avenue for developing future flow models that are more efficient, expansive, and precise.

References

[1] J. Huang, W. Zou, J. Zhu, and Z. Zhu, "Optical Flow Based Real-time Moving Object Detection in Unconstrained Scenes," July 2018. arXiv:1807.04890 [cs].

[2] G. C. H. E. de Croon, C. De Wagter, and T. Seidl, "Enhancing optical-flow-based control by learning visual appearance cues for flying robots," *Nature Machine Intelligence*, vol. 3, pp. 33–41, Jan. 2021. Number: 1 Publisher: Nature Publishing Group.

[3] S. Baker and I. Matthews, "Lucas-Kanade 20 Years On: A Unifying Framework," *International Journal of Computer Vision*, vol. 56, pp. 221–255, Feb. 2004.

[4] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, Aug. 1981.

[5] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks," Dec. 2016. arXiv:1612.01925 [cs].

[6] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning Optical Flow with Convolutional Networks," May 2015. arXiv:1504.06852 [cs].

[7] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume," June 2018. arXiv:1709.02371 [cs].

[8] Z. Teed and J. Deng, "RAFT: Recurrent All-Pairs Field Transforms for Optical Flow," Aug. 2020. arXiv:2003.12039 [cs].

[9] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, "Event-based Vision: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, pp. 154–180, Jan. 2022. arXiv:1904.08405 [cs].

[10] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras," in *Robotics: Science and Systems XIV*, June 2018. arXiv:1802.06898 [cs].

[11] M. Gehrig, M. Millhausler, D. Gehrig, and D. Scaramuzza, "E-RAFT: Dense Optical Flow from Event Cameras," in *2021 International Conference on 3D Vision (3DV)*, (London, United Kingdom), pp. 197–206, IEEE, Dec. 2021.

[12] Y. Wu, F. Paredes-Vallés, and G. C. H. E. de Croon, "Rethinking Event-based Optical Flow: Iterative Deblurring as an Alternative to Correlation Volumes," Mar. 2023. arXiv:2211.13726 [cs].

[13] G. Gallego, H. Rebecq, and D. Scaramuzza, "A Unifying Contrast Maximization Framework for Event Cameras, with Applications to Motion, Depth, and Optical Flow Estimation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3867–3876, June 2018. arXiv:1804.01306 [cs].

[14] M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza, "DSEC: A Stereo Event Camera Dataset for Driving Scenarios," *IEEE Robotics and Automation Letters*, vol. 6, pp. 4947–4954, July 2021.

[15] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The Multivehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception," *IEEE Robotics and Automation Letters*, vol. 3, pp. 2032–2039, July 2018. Conference Name: IEEE Robotics and Automation Letters.

[16] G. Bird and M. E. Polivoda, "Backpropagation Through Time For Networks With Long-Term Dependencies," Apr. 2021. arXiv:2103.15589 [cs].

[17] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised Event-based Learning of Optical Flow, Depth, and Egomotion," Dec. 2018. arXiv:1812.08156 [cs].

[18] J. Hagenaars, F. Paredes-Valles, and G. de Croon, "Self-Supervised Learning of Event-Based Optical Flow with Spiking Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 34, pp. 7167–7179, Curran Associates, Inc., 2021.

[19] S. Bai, J. Z. Kolter, and V. Koltun, "Deep Equilibrium Models," Oct. 2019. arXiv:1909.01377 [cs, stat].

[20] S. Bai, Z. Geng, Y. Savani, and J. Z. Kolter, "Deep Equilibrium Optical Flow Estimation," Apr. 2022. arXiv:2204.08442 [cs].

[21] B. Nguyen and L. Mauch, "Efficient Training of Deep Equilibrium Models," Apr. 2023. arXiv:2304.11663 [cs].

[22] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, pp. 43–77, Feb. 1994.

[23] H. Liu, G. Chen, S. Qu, Y. Zhang, Z. Li, A. Knoll, and C. Jiang, "TMA: Temporal Motion Aggregation for Event-based Optical Flow," Aug. 2023. arXiv:2303.11629 [cs].

[24] D. Lin, H. Ye, and Z. Zhang, "Explicit Superlinear Convergence Rates of Broyden's Methods in Nonlinear Equations," Sept. 2022. arXiv:2109.01974 [math].

[25] H. F. Walker and P. Ni, "Anderson Acceleration for Fixed-Point Iterations," *SIAM Journal on Numerical Analysis*, vol. 49, pp. 1715–1735, Jan. 2011.

[26] Z. Geng, X.-Y. Zhang, S. Bai, Y. Wang, and Z. Lin, "On Training Implicit Models," Jan. 2022. arXiv:2111.05177 [cs].

[27] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural Ordinary Differential Equations," Dec. 2019. arXiv:1806.07366 [cs, stat].

[28] S. W. Fung, H. Heaton, Q. Li, D. McKenzie, S. Osher, and W. Yin, "JFB: Jacobian-Free Backpropagation for Implicit Networks," Dec. 2021. arXiv:2103.12803 [cs].

[29] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Jan. 2017. arXiv:1412.6980 [cs].

[30] L. N. Smith and N. Topin, "Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates," May 2018. arXiv:1708.07120 [cs, stat].

[31] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," Dec. 2019. arXiv:1912.01703 [cs, stat].

[32] S. Shiba, Y. Aoki, and G. Gallego, "Secrets of Event-Based Optical Flow," July 2022. arXiv:2207.10022 [cs].

Appendix A

Optical Flow Coding Scheme

The visualizations of the optical flow used in the figures are created by mapping the magnitude and direction of optical flow vectors into brightness and color hue, respectively, as shown in Figure 6.

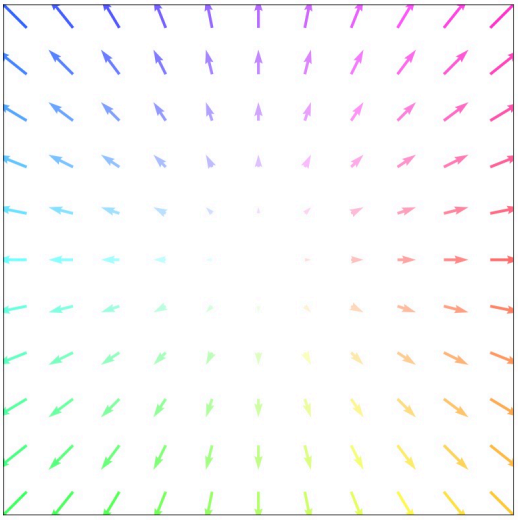


Figure 6: Optical flow coding scheme where the magnitude of the optical flow is encoded in brightness and the direction in color hue.

Appendix B

Additional Results on Self-Supervised Learning

Figure 7: Extracted from sequence *indoor_flying_1*. Ground truth flow is on the left, and optical flow prediction is on the right.

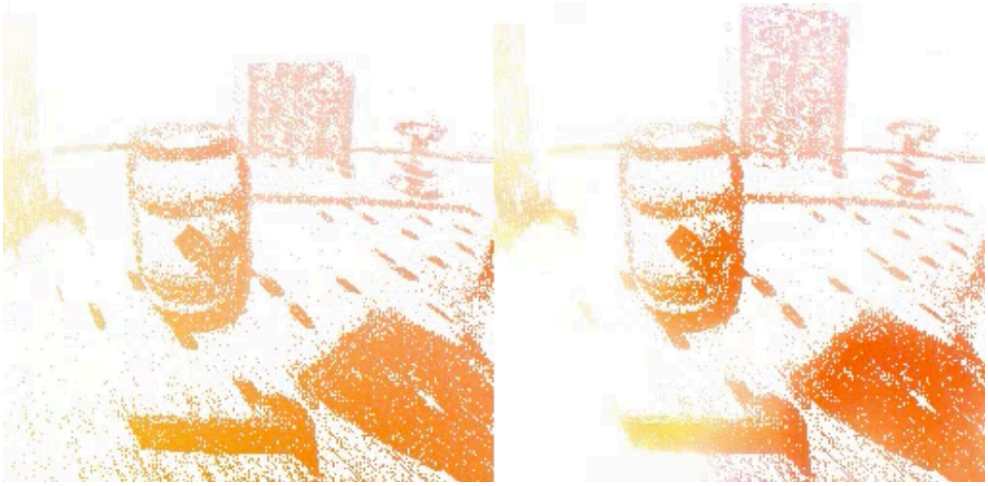


Figure 8: Extracted from sequence *indoor_flying_1*. Ground truth flow is on the left, and optical flow prediction is on the right.

Figure 9: Extracted from sequence *indoor_flying_1*. Ground truth flow is on the left, and optical flow prediction is on the right.

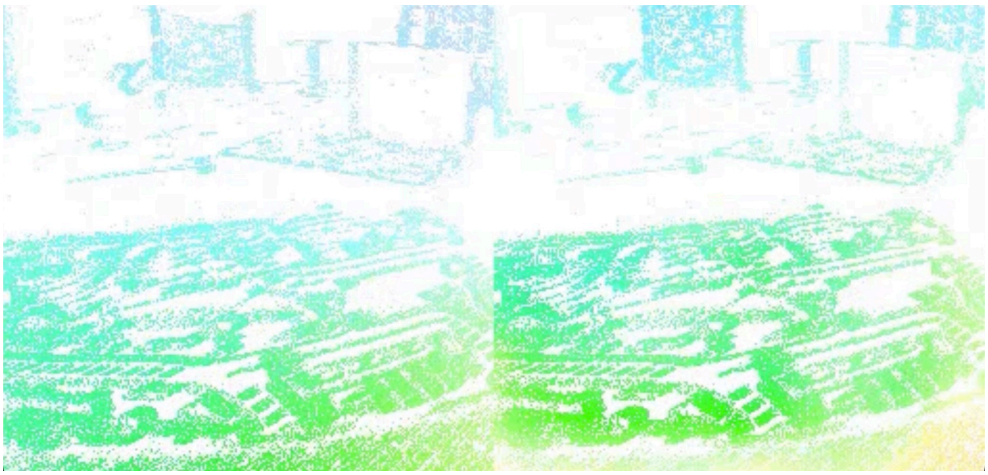


Figure 10: Extracted from sequence *indoor_flying_2*. Ground truth flow is on the left, and optical flow prediction is on the right.

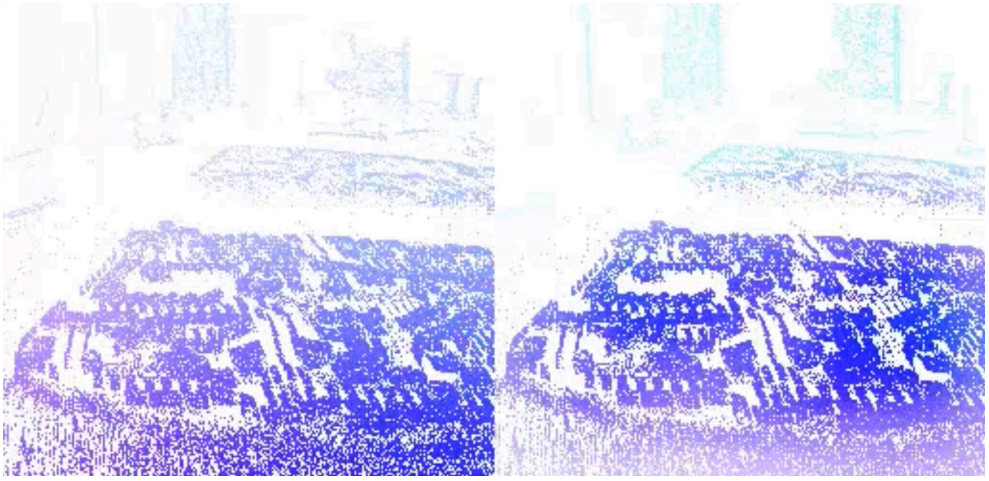


Figure 11: Extracted from sequence *indoor_flying_2*. Ground truth flow is on the left, and optical flow prediction is on the right.

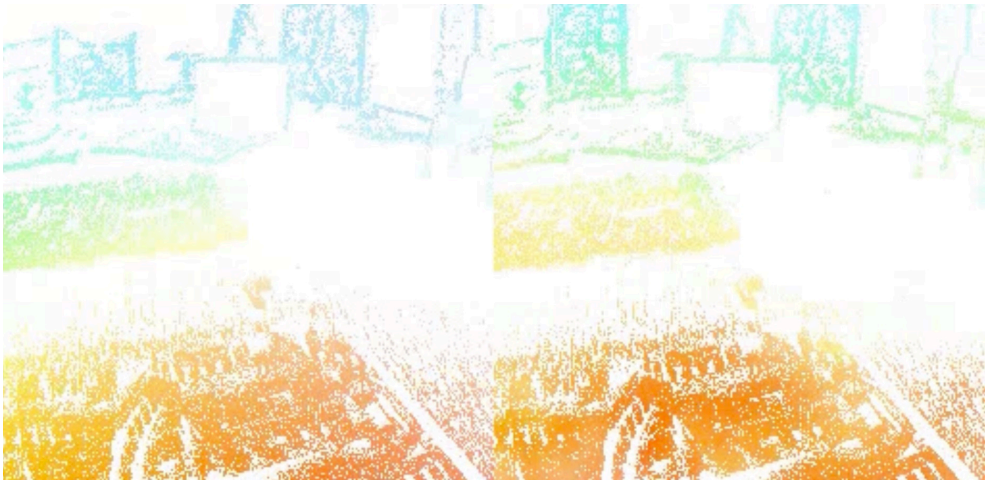


Figure 12: Extracted from sequence *indoor_flying_2*. Ground truth flow is on the left, and optical flow prediction is on the right.

Figure 13: Extracted from sequence *indoor_flying_3*. Ground truth flow is on the left, and optical flow prediction is on the right.

Figure 14: Extracted from sequence *indoor_flying_3*. Ground truth flow is on the left, and optical flow prediction is on the right.

Figure 15: Extracted from sequence *indoor_flying_3*. Ground truth flow is on the left, and optical flow prediction is on the right.

Part II

Preliminary Analysis

3

Literature Review

This chapter comprises a review of prior research and literature relevant to the focal area of this study. Starting with Section 3.1 where the working principles of event-based cameras are explained. An investigation is also carried out to list the currently available event-based datasets. Following that is Section 3.2, which covers the traditional frame-based optical flow estimation methods and delves into the current methods for computing optical flow from event-based data. Section 3.3 gives an overview of how self-supervised learning can be used to eradicate the need for ground truth labels and emergent methods of applying it to the problem of event-based optical flow. Finally, Section 3.4 concludes the chapter with an investigation into the working of Deep Equilibrium Models and their applicability to the problems touched by this research.

By exploring the landscape of existing work, we aim to contextualize our research, identify gaps, and highlight the contributions and limitations of previous studies. This critical analysis not only provides a solid foundation for our current investigation but also serves as a compass, guiding the trajectory of our exploration within this field.

3.1. Event-Based Data

Event cameras represent a departure from the frame-based paradigm of conventional cameras, offering advantages in terms of speed, efficiency, and adaptability to fast-changing environments. They're particularly well-suited for applications where quick and precise visual information is crucial due to their high dynamic range and robustness to motion blur.

Instead of capturing discrete frames at a certain frame rate, each pixel in an event camera operates independently, recording light intensity and triggering an "event" every time the change in intensity passes a certain threshold. This event is usually defined as $\mathbf{e}_k = (x_k, y_k, t_k, p_k)$ with x_k and y_k being the coordinates of the pixel triggering the event, t_k being the timestamp and p_k being the signed intensity change or polarity.

3.1.1. Datasets

In order to develop and train algorithms that operate on this novel data format, vast amounts of data are needed. This data needs to not only contain the event streams captured by the event cameras, but also ground truth that can be used to train and evaluate these algorithms on problems such as object detection, optical flow, and others.

Event-based Data for Pose Estimation, Visual Odometry, and SLAM

The Event-based Data for Pose Estimation, Visual Odometry, and SLAM is a work by Mueggler et al. [11]. It presents one of the first event-based datasets, consisting of multiple sequences of objects undergoing simple translational and rotational movement captured by the DAVIS240C event camera with a resolution of 240 by 180 pixels. This effort has allowed scientists to explore the possibilities of applying neural networks to event-based data. The dataset also includes simultaneously captured grayscale frames of the same resolution but lacks optical flow ground truth which limits the possibility of applying supervised learning algorithms. Nevertheless, it can be used to provide useful qualitative metrics due to its simple structure.

MVSEC
The Multi Vehicle Stereo Event Camera [12] dataset is one of the first public datasets to provide event-based data along with optical flow ground truth. Utilizing the DAVIS m346B event camera, it provides a continuous event stream with a resolution of 346 by 260 pixels. Capturing event sequences in both indoor and outdoor conditions during day and night settings, it fuses lidar, IMU, and GPS data to also provide sparse ground truth labels for the optical flow

DSEC
The Stereo Event Camera Dataset for Driving Scenarios [13] is a more recent attempt at providing a comprehensive event dataset, capturing streams from two event cameras in a stereo setup with a resolution of 640 by 480 pixels. The effort consists of 53 driving sequences exposed to different varieties of illumination while providing sparse but accurate ground truth for optical flow estimation.

BlinkFLow
Blinkflow [14] is one of the most recently released datasets. And it substantially deviates from the standard procedure of collecting events using event cameras in real-world conditions. Instead, Blinkflow employs an event data simulator, which is used to generate a large-scale, diverse dataset containing more than 3362 training sequences at a 640 by 480 pixels resolution. Having such an abundant dataset is key in the exploration of the applicability of neural networks to event-based data.

3.2. Optical Flow Estimation
Frame-based Methods
The estimation of optical flow stands as a cornerstone problem in the area of computer vision, spanning multiple decades of investigation. In the early 1980s, the pioneering work of Horn and Schunck [4] laid the groundwork for what has been the standard algorithm for determining optical flow. Their seminal paper introduced a method that computed optical flow by minimizing an energy function based on two critical assumptions:

1. **Brightness constancy**: the brightness intensity of pixels remains constant over time.
2. **Spatial smoothness**: neighboring pixels tend to have similar motion patterns.

which are used to formulate the terms in the energy function. This energy function is then minimized through iterative procedures such as gradient descent to produce a dense optical flow field, representing the motion vectors for each pixel in the image.

While the Horn-Schunck method was groundbreaking, it had limitations, particularly in handling large displacements, occlusions, and complex motion patterns. Therefore, in the following years, researchers explored various approaches to address these limitations, such as the use of pyramidal structures [15]. The concept involves creating a pyramid of images by successively downsampling the original image. Each pyramid level represents the image at a different scale, where higher levels correspond to lower resolutions. This pyramid allows optical flow algorithms to operate at multiple scales, enabling them to handle different levels of detail and motion complexities within an image sequence. Still, the added complexity significantly increased the latency of computing flow using that approach, limiting its use in real-time applications.

More recently, with the advent and rapid adoption of deep learning methods in the past decade, researchers began investigating ways of applying neural networks to the problem of optical flow estimation. This posed a paradigm shift from the traditional hand-crafted optimization framework by formulating it as an end-to-end learning task with the network directly predicting the flow from a set of frames.

FlowNet [16] was the first to adopt the formulation of a supervised CNN capable of solving the optical flow estimation problem. To do so, it employs three key principles:

1. **Feature Extraction**: The network extracts intricate details and hierarchical representations from the frames through convolutional layers, capturing essential information for understanding motion.
2. **Correlation Calculation**: Using these extracted features, FlowNet computes correlations between the feature maps, enabling the identification of matching patterns and potential motion across the frames.

3. **Refinement**: FlowNet utilizes ”upconvolutional” layers that upsample the low-resolution flow to provide dense per-pixel predictions.

FlowNet [16] and its successor FlowNet 2.0 [17] achieved performance matching that of state-of-the-art energy minimization methods while achieving much lower inference times, proving the potential of this data-driven approach.

PWC-Net [5] further builds on these ideas by introducing three crucial improvements. Firstly, it employs the pyramidal structures, common in traditional methods, allowing it to generate feature representation at multiple scales. It also introduces a ”warping” layer, which warps one feature map towards the other based on the predicted flow, helping with the alignment of corresponding pixels. Moreover, it constructs a cost volume by comparing features between the two images at multiple scales. This volume represents the similarity or dissimilarity between the features of corresponding pixels. Finally, based on the refined cost volume information, the network predicts the optical flow, representing the motion between the two input frames. Using these ideas, PWC-Net outperforms FlowNet2 while being 17 times smaller in size.

Another architecture that introduces a monumental shift is that of RAFT [6]. While networks such as FlowNet [16] and PWC-Net [5] process their input in an entirely sequential fashion, RAFT makes use of a recurrent unit to refine its flow predictions iteratively. To do so, it builds cost volumes containing correlations between all pairs of pixels, which are then fed through the recurrent unit to update the flow estimate incrementally. RAFT's recurrent unit helps enforce temporal consistency. By iteratively refining flow estimation through recurrent updates, it encourages the model to generate flow fields that are consistent across time steps. This leads to smoother and more temporally coherent flow predictions, allowing the architecture to outperform its predecessors substantially.

Event-based Methods

Event-based cameras have brought a radical new way of capturing motion information through asynchronous event streams. This format presents an untapped source of potential that could unlock many new possibilities in many areas, such as computer vision and robotics, among many others. But this novel format also poses an extraordinary challenge since both traditional and deep learning algorithms cannot be directly applied. But thankfully, due to the traction that event cameras have created, substantial effort has recently been put into developing algorithms that could deal with and exploit their data structure.

Zhu et al.[18] propose an encoder-decoder architecture that utilizes a novel event representation. The purpose of this representation is to discretize the continuous event stream into a voxel grid that compresses the spatial and temporal information of all events into a single representation that existing learning methods can utilize.

Starting with a set of N events (x_i, y_i, t_i, p_i) where $i \in [1, N]$ and B bins to discretize the time-domain, each event is bilinearly interpolated both temporally and spatially using Equation 3.2.

$$t_i^* = (B-1)\,(t_i - t_0)\,/\,(t_N - t_1) \tag{3.1}$$

$$V(x, y, t) = \sum_i p_i k_b\,(x - x_i)\,k_b\,(y - y_i)\,k_b\,(t - t_i^*) \tag{3.2}$$

$$k_b(a) = max(0, 1 - |a|) \tag{3.3}$$

where $k_b(a)$ represents the bilinear kernel. The results of this interpolation process are then accumulated into a voxel grid tensor of size BxHxW, allowing for each bin to be treated as a separate 2D image. The produced bins are then fed through a convolutional encoder-decoder network to estimate the optical flow encoded in the events.

E-RAFT [9] applies the RAFT [6] architecture on event data by utilizing the exact event representation described by Zhu et al.[18]. The method achieves state-of-the-art performance, showcasing the superiority of applying recurrent models on data with a strong temporal nature.

Liu et al. [19] further improve upon E-RAFT [9] by making use of three additional components in their TMA (Temporal Motion Aggregation) architecture. Firstly, they introduce an event-splitting strategy that divides an event stream into multiple segments that are encoded into feature maps, which are used to

compute temporally dense correlation volumes. Secondly, these volumes, along with the corresponding flow, are processed using a linear look-up module to encode motion features. Finally, instead of just concatenating these motion features to produce a final estimate, an attention module is used to emphasize motion patterns that are consistent with each other. This allows the method to outperform E-RAFT on the DSEC [13] benchmark.

However, the computation and storage of the correlation volumes used by both E-RAFT and TMA greatly increase their memory consumption and inference time, diminishing the possibility of deploying the architectures on a real-time platform, especially as the input resolution increases.

IDNet [10] targets this problem by directly eliminating the correlation volumes and instead observing that the motion of objects is directly encoded in the blur present in the raw events. By integrating the idea of motion compensation [20], IDNet also refines its estimate of the optical flow by iteratively ”deblurring” the event representations and processing them using a recurrent unit. This allows IDNet to achieve exceptional results on the competitive DSEC [13] benchmark, indicating the power of introducing inductive bias when designing neural network architectures.

3.3. Self-Supervised Learning

Despite the success of these methods in utilizing the novel event data format to obtain accurate predictions of optical flow, most of them still require substantial amounts of annotated ground truth data in order to train their underlying neural networks. Unfortunately, the process of creating such datasets is extremely laborious, subject to errors, and time-consuming.

Supervised methods such as [19, 9, 10] have so far focused on the two benchmark datasets - MVSEC [12] and DSEC [13], using the difference between their predictions and the ground truth optical flow as a loss metric. However, both datasets have their optical flow ground truth as a sparse structure and are provided at a low frequency, further aggravating the training process.

Another prominent method of training neural networks on computer vision tasks such as optical flow estimation [21, 22] is via the use of self-supervised learning. Self-supervised learning is a machine learning paradigm where a model learns to understand patterns, features, or representations from the input data itself without relying on external labels or annotations. Instead of using manually labeled data, self-supervised learning tasks generate labels or supervisory signals from the input data. As such, this training paradigm holds enormous potential in not only removing the need for manual labeling but also allowing models to be trained on much bigger datasets that do not contain ground truth data.

EV-FlowNet [8] formulates the training regime as a self-supervised problem using only event data and a set of corresponding grayscale images generated from the same camera. To do so, it uses the predicted optical flow and two grayscale images corresponding to the beginning and end of the optical flow interval. It then warps the second image back to the first one with the idea that as the optical flow prediction converges to the true optical flow, the warped and the original images will become identical. To measure their dissimilarity, the photometric loss between the warped and original images is computed, which, along with a smoothness term, is used as a supervisory signal.

The method, while successful, still requires a set of corresponding grayscale images on top of the event data to generate the supervisory signal, which requires a complicated synchronization process. Therefore, substantial research has been put into algorithms that employ only the raw event data to generate that supervisory signal, and the most efficient framework that does so is that of contrast maximization, which can be subdivided into two categories: spatial and temporal contrast maximization.

3.3.1. Spatial Contrast Maximization

Apart from the general requirement of differentiability, a loss function should, first and foremost, provide a clear distinction between good and bad predictions made by the neural network. In the case of spatial contrast maximization, this can be achieved by utilizing the concept of motion compensation as introduced by [10, 20]. Starting with a set of events $E = \{x_k\ y_k\ t_k\ p_k\}$ where $k \in [1, N]$, the predicted optical flow from the model is used to warp the events to a reference time t_{ref}, generating a set of warped events E' using the following warp:

$$\boldsymbol{x'_k} = \boldsymbol{x_k} + (t_k - t_{ref})\boldsymbol{v}(\boldsymbol{x_k}) \tag{3.4}$$

where $\boldsymbol{v(x_k)}$ is the predicted optical flow vector. The warped events are then aggregated to an image of warped events (IWE) using

$$I(\boldsymbol{x}) = \sum_{k=1}^{N} g(\boldsymbol{x} - \boldsymbol{x'_k}) \tag{3.5}$$

with each pixel accumulating the warped events that fall within it using the function g. When the data contains floating point coordinates, for example, when considering rectified coordinates, the function g represents the bilinear kernel [10] while for integer coordinates, it can be replaced by a Dirac delta.

Under a perfect optical flow prediction, this IWE would have a perfect edge alignment, resulting in sharp edges and no motion blur as seen in Figure 3.1.

(a) Iteration $it = 1$ (b) $it = 2$ (c) $it = 3$ (d) $it = 5$

Figure 3.1: Progression of event warping as the optical flow prediction improves[23].

Now, the question is how to robustly measure this alignment arises. Gallego et al. [23] propose a set of twenty loss functions that aim to do just that. Starting with the IWE, they apply statistical measures as well as image processing techniques and assess the performance of all of them.

The first of these functions is the variance of the IWE, also known as the *contrast* [23, 24], and is computed using:

$$Var(I(\boldsymbol{x})) \doteq \frac{1}{\Omega} \int_{\Omega} (I(\boldsymbol{x} - \mu_I))^2 d\boldsymbol{x} \tag{3.6}$$

with μ_I being the mean of all the pixels in the IWE. In statistics, the variance quantifies the spread or dispersion of a set of data points around their mean or average value, giving an idea of how much the values in a dataset differ from the mean. Looking from the point of an image, a high variance in an image implies that there are significant differences in pixel intensities, which often correspond to areas with sharp transitions or edges. On the other hand, a low variance suggests more uniform regions or areas with less contrast and fewer intensity variations. Therefore, by maximizing that metric, the alignment of edges is also maximized.

The magnitude of the image gradient is another metric that is able to capture this effect [25], calculated via:

$$\|\nabla I\|_{\Omega}^2 \doteq \int_{\Omega} \|\nabla I(\mathbf{x})\|^2 d\mathbf{x} \tag{3.7}$$

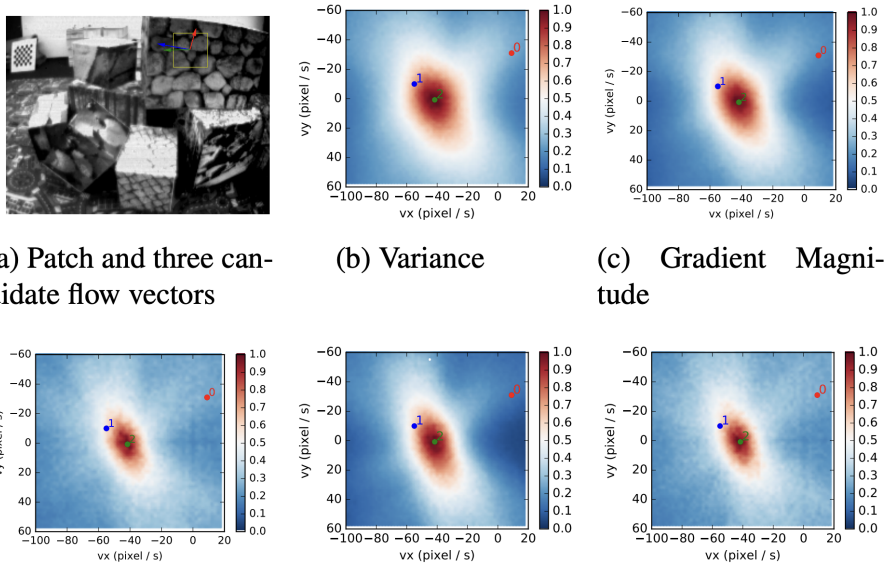
with $\nabla I = (I_x, I_y)^T$ being the gradient vector. It is often used to detect edges and boundaries within the image. It identifies areas where there's a significant change in intensity or color, which typically indicates an edge or a transition between different objects or regions in the image. When that magnitude is maximized, it indicates that edge alignment is also maximized.

The magnitude of the image Laplacian proves to be an efficient metric as well as noted by [23] and is computed using:

$$\|\Delta I\|_{\Omega}^2 \doteq \|I_{xx} + I_{yy}\|_{\Omega}^2 \tag{3.8}$$

The Laplacian operator highlights regions in the image where the intensity changes rapidly. It is particularly effective in detecting edges and fine details that might not be easily discernible in the original image. Maximizing this metric again leads to a positive effect in the alignment of edges in the IWE.

Despite the similarity between these metrics, they also exhibit different properties mathematically, which could ease or deteriorate the process of learning how to correctly estimate the optical flow. Figure 3.2 presents the findings by Gallego et al. [23] regarding the aforementioned metrics along with two other. It depicts a plot of the different losses with respect to the optical flow vector's components. In this case, the narrower the peak, the better that metric is at allowing the network to learn, with the magnitude and variance of the Laplacian coming on top.



(a) Patch and three candidate flow vectors (b) Variance (c) Gradient Magnitude

(d) Laplacian Magnitude (e) Variance of Gradient Magnitude. (f) Variance of Laplacian.

Figure 3.2: Loss metrics as a function of optical flow parameters, indicating that some metrics are better suited for the estimation of optical flow [23]

Shiba et al. [25] note that when optimizing the variance of the IWE, strong overfitting behavior can be observed, which prevents the network from learning actual patterns in the data that lead to correct and physically possible flow. To prevent that, they introduce a multi-reference objective function, which warps the IWE to multiple reference times, instead of just one, acting as a regularizing term.

3.3.2. Temporal Contrast Maximization

Temporal contrast maximization aims to achieve the same goal as the methods described previously but by looking at the problem from a different standpoint. Zhu et al. [8] note that another way of maximizing the contrast of the IWE is by minimizing its per-pixel, per-polarity average timestamp, which is summarized by:

$$T_{p'}\left(\mathbf{x};\mathbf{u} \mid t_{\text{ref}}\right) = \frac{\sum_j \kappa\left(x - x'_j\right)\kappa\left(y - y'_j\right)t_j}{\sum_j \kappa\left(x - x'_j\right)\kappa\left(y - y'_j\right) + \epsilon} \tag{3.9}$$

$$\kappa(a) = max(0, 1 - |a|) \tag{3.10}$$

$$j = \{i \mid p_i = p'\}, \quad p' \in \{+,-\}, \quad \epsilon \approx 0 \tag{3.11}$$

The loss is then computed by taking the sum of both images generated using positive and negative polarity as in:

$$\mathcal{L}_{\text{contrast}}\ (t_{ref}) = \sum_x \sum_y T_+\ (x, y \mid t_{ref})^2 + T_-\ (x, y \mid t_{ref})^2 \tag{3.12}$$

Unfortunately, with its current formulation, the method suffers from scaling problems, namely that by using a single reference time for the warp, the events are scaled by $(t' - t_i)$. This leads to events with timestamps closer to t' having virtually no contribution while events further away from t' have a disproportionately larger contribution during backpropagation. To solve this problem, the authors compute the loss with respect to both the beginning and the ending of the event sequence, denoted by timestamps t_0 and t_{N-1} respectively, leading to:

$$\mathcal{L}_{\text{contrast}} = \mathcal{L}_{\text{contrast}}\ (t_0) + \mathcal{L}_{\text{contrast}}\ (t_{N-1}) \tag{3.13}$$

However, Hagenaars et al. [26] find that even using two reference times to compute the contrast maximization loss could lead to learning problems due to the non-convex nature of the function. To alleviate this problem, they scale the loss described in Equation 3.14 by the sum of pixels with at least one warped event to convexify the metric, resulting in:

$$\mathcal{L}_{\text{contrast}}\ (t_{ref}) = \frac{\sum_x \sum_y T_+\ (x, y \mid t_{ref})^2 + T_-\ (x, y \mid t_{ref})^2}{\sum_x \sum_y n(x' > 0) + \epsilon} \tag{3.14}$$

with $n(x')$ counting the per-pixel events of the IWE.

In order to introduce a regularizing term that steers the estimated flow in a physically possible direction, apart from the contrast maximization loss, a smoothing loss is also incorporated as suggested by [18, 26, 27, 28]. This smoothing term takes the form of:

$$\mathcal{L}_{\text{smooth}} = \sum_{\vec{x}} \sum_{\vec{y} \in \mathcal{N}(\vec{x})} \rho(u(\vec{x}) - u(\vec{y})) + \rho(v(\vec{x}) - v(\vec{y})) \tag{3.15}$$

$$\rho(x) = \sqrt{x^2 + \epsilon^2} \tag{3.16}$$

with $\mathcal{N}(x, y)$ being the set of neighbors around (x, y) and ϵ is a small constant added to prevent division by zero and/or numerical under or overflow.

3.4. Adaptive Inference

IDNet employs the idea of using iterative refinement to improve its optical flow estimation. It uses its current optical flow estimation to deblur the event sequence used to generate it, which is then fed again into its recurrent module to update the flow. But currently that iteration number is manually chosen and fixed, which comes with negative consequences.

Firstly, the number of forward passes stays constant regardless of the complexity of the scene, which adds undesired latency in cases where the motion patterns are simple to estimate. For example, considering a scene with no motion, the output of the network would be a trivial zero flow field. Despite that, the network would still have to go through all iterations to arrive at that result, wasting computing power and time.

On the other hand, it is also possible that the scene at hand contains various objects with different movement directions and magnitudes. This would produce a complex optical flow map that might require a greater number of iterations to converge to. Yet with the current schemes, the iterations will be cut short, leading to worse predictions and decreased accuracy.

Therefore, adding a method of automatically adjusting the number of iterations performed by the network that is dependent on the complexity of the motion field in the sequence of events can prove to be a great improvement not only in the inference speed of IDNet but also in its accuracy. A promising way to look at this problem is through the lens of Deep Equilibrium Models (DEQ) [29].

3.4.1. Deep Equilibrium Models

Traditional optical flow estimation methods [4] rely on an energy minimization principle that is iteratively optimized until a stable flow field is produced. While recurrent networks such as IDNet mimic this principle by iteratively refining their estimate, this is only carried out for a certain number of steps, which also comes with major drawbacks. Mainly that training such recurrent models involves back-propagation-through-time (BPTT), which comes with a significant time and memory burden. But what if this iterative procedure could be replaced with an architecture that can immediately compute this stable solution and without the need for BPTT?

This is exactly the idea behind Deep Equilibrium Models. By re-formulating the architecture as an infinitely deep implicit layer, the outputs of the model become the fixed points of this new implicit layer, which can be directly solved for. To explain this principle, consider a recurrent unit of the form:

$$z_{i+1} = f_\theta(Wz_i + Ux + b) \tag{3.17}$$

As the number of iterations goes to infinity, the output of this model could take one of three possible paths: convergence, divergence, or instability in the form of oscillations. In the case where convergence occurs, the output z^* can be considered as the solution of a fixed-point iteration scheme as in:

$$z^* = f_\theta(Wz^* + Ux + b) = f_\theta(z^*, x) \tag{3.18}$$

Therefore, to find that equilibrium state, a fixed-point solver such as Newton's or Broyden's method [29] can be utilized. If you were to train a network using the current formulation, this would require backpropagating through all the iterations of the fixed-point solver, which would leave us with the same problem that the BPTT presented, showing no benefits to using this fixed-point formulation. Luckily, DEQ models can compute derivatives directly through the final fixed point, z^*, making use of the Implicit Differentiation Theorem [29]. This allows DEQs to compute the derivatives with respect to any parameter in the network using the following equation [29]:

$$\frac{\partial \ell}{\partial(\cdot)} = -\frac{\partial \ell}{\partial \mathbf{z}^\star}\left(J_{g_\theta}^{-1}\big|_{\mathbf{z}^\star}\right)\frac{\partial f_\theta\left(\mathbf{z}^\star;\mathbf{x}\right)}{\partial(\cdot)} = -\frac{\partial \ell}{\partial h}\frac{\partial h}{\partial \mathbf{z}^\star}\left(J_{g_\theta}^{-1}\big|_{\mathbf{z}^\star}\right)\frac{\partial f_\theta\left(\mathbf{z}^\star;\mathbf{x}\right)}{\partial(\cdot)} \tag{3.19}$$

The most critical implication of this step is that the need to store intermediate hidden states, which is at the heart of BPTT, is rendered obsolete. This allows DEQs to compute the gradient by using only the Jacobian evaluated at the equilibrium point, which decreases the memory needed during training massively.

Bai et al. [30] apply this strategy directly to the problem of estimating optical flow. Since the technique is not based on any specific model, it can be integrated into already existing networks such as RAFT [6]. The authors note that training RAFT as a DEQ module allowed them to achieve superior results at the same computational cost and at a fraction of the memory footprint, validating the efficacy of DEQ models.

Yet computing the gradient during backpropagation still requires the calculation of the expensive inverse Jacobian matrix $J_{g_\theta}^{-1}$, which becomes intractable for bigger problems. Early works [29] suggest computing this matrix iteratively by making use of Broyden's method, which allows for more efficient computation, especially when integrated with already existing automatic differentiation packages. Instead of relying on this iterative method again, Nguyen et al. [31] approximate the matrix using the inverse Jacobian computed during the forward pass. This allows for more than a 2x speed-up without sacrificing accuracy.

4

Preliminary Results

This chapter will discuss the preliminary results of applying the self-supervised learning methods discussed in Chapter 3. This will be done in Section 4.1 and Section 4.2, where the implementation and results of the spatial and contrast maximization principles will be presented, respectively.

4.1. Spatial Contrast Maximization

The first step in implementing the spatial contrast maximization framework into IDNet starts with utilizing the deblurred image of warped events that is already computed in the iterative structure of the network. As mentioned in Chapter 3, under perfect estimation of the optical flow, the IWE's contrast is maximized, resulting in the recovery of sharp edges in the image. Experiments were conducted on the DSEC [13] dataset and specifically on the *zurich_city_01_a* sequence to test the objective functions, keeping the same model hyperparameters that IDNet uses during its supervised training regime - 15 bins collecting events in an interval of 100 [ms]. Using that sequence, the model was trained until convergence using Adam with a learning rate of 1×10^{-4} and batch size of 5.

The first metric used to measure this alignment quantitatively is the variance of the IWE as described in Equation 3.6. This metric aims to do so by measuring the difference in pixel intensities, with higher results corresponding to bigger contrast. Therefore, minimizing the negative of that function results in IWEs with higher contrast. By looking at the loss progression as presented in Figure 4.1, we can conclude that the model is indeed learning.

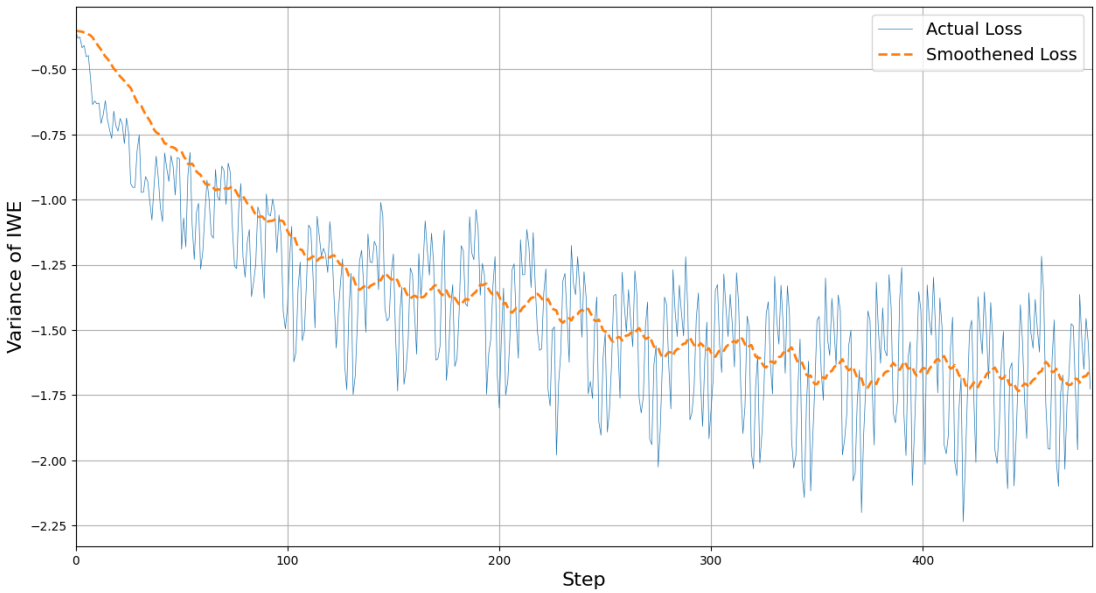


Figure 4.1: Loss progression during training, using the Variance of the IWE.

The loss monotonically decreases and plateaus towards the end, oscillating around an average value and indicating convergence has occurred. This could lead to the belief that the model is indeed learning to predict the optical flow correctly. However, the training loss does not tell the whole picture when using self-supervised learning. Therefore, it is of utmost importance to verify the validity of the flow by directly inspecting it, along with the image of warped events. These results are presented in Figure 4.2.

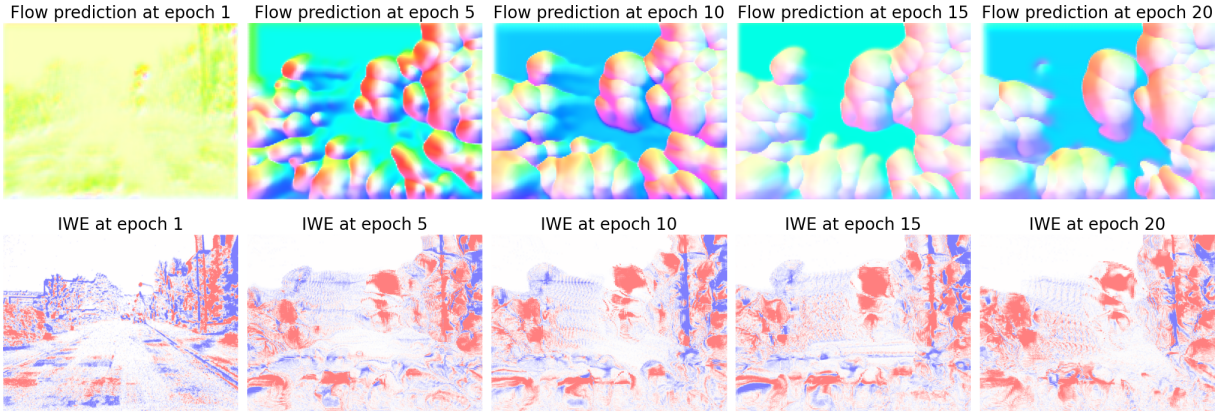


Figure 4.2: Optical flow prediction and corresponding IWE recorded during training when optimizing the Variance of the IWE.

As seen in the figure, while the network is initially able to at least recognize patterns in the flow, the flow predictions steadily devolve as the training progresses. This causes the formation of ”bubble”-like regions that form similar patterns when used to produce the IWE. Naturally, the question of how these non-physical patterns emerge in the flow arises. The answer to that lies precisely in the aforementioned bubble-like regions. These regions present very abrupt changes in intensity compared to their surroundings, which raises the variance of the whole image. Therefore, even though the estimated flow is not correct, it is optimal according to the loss function. This can be considered as an example of overfitting where the underlying model does not learn the correct patterns in the data, but instead blindly tries to decrease the loss function as instructed.

The next loss that was tested is the magnitude of the gradient of the IWE as shown in Equation 3.7. The idea behind it being that the more pronounced the edges are, the higher its value gets, thus maximizing contrast. Its progression can be followed in Figure 4.3.

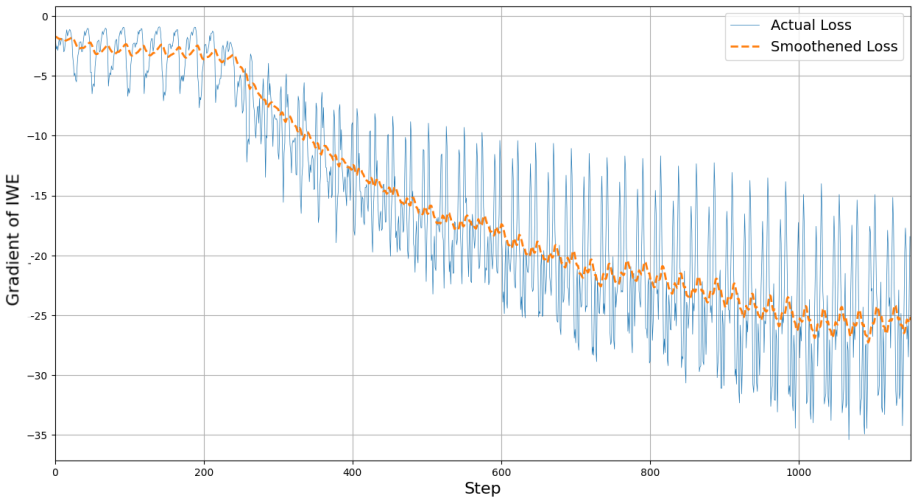


Figure 4.3: Loss progression during training, using the Gradient of the IWE.

Training IDNet using the gradient of the IWE as its supervisory signal takes more time to converge

compared to using the variance of the IWE. One thing that is especially noticeable is that initially, the loss oscillates without much improvement until it suddenly takes off around step 200. This could indicate that the gradient loss poses a more complicated landscape containing multiple local minima, making it harder for the optimization algorithm to optimize it. Nevertheless, it reaches convergence towards the end, hinting that the network is learning again.

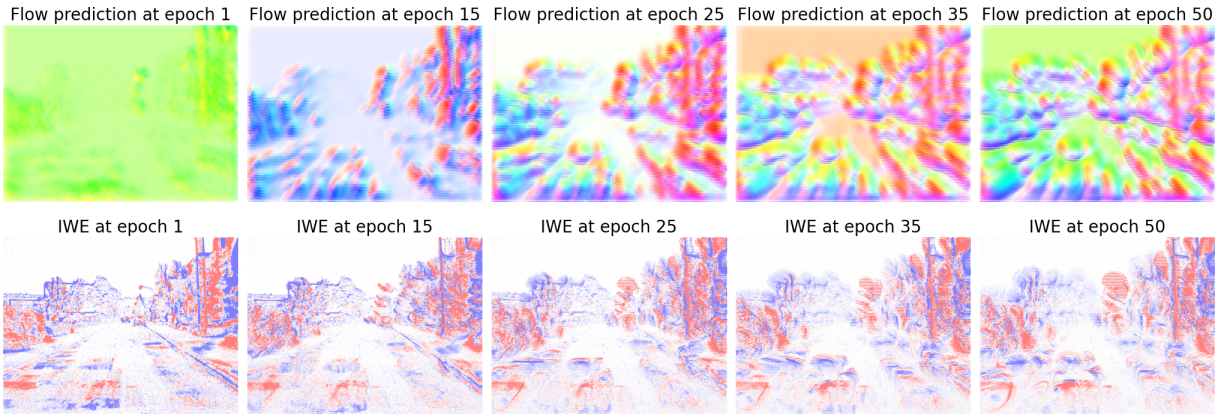


Figure 4.4: Optical flow prediction and corresponding IWE recorded during training when optimizing the Gradient of the IWE.

But after inspecting the flow and corresponding IWEs in Figure 4.4, we are left at the same point as previously, with a network that is able to optimize its loss without learning the true patterns in the flow. This time, instead of producing the bubble-like structures present in Figure 4.2, the network generates patterns of horizontal stripes with sharp transitions in contrast, which effectively maximize the gradient of the IWE but produce inaccurate optical flow predictions.

The final loss that was investigated is the Laplacian of the IWE as shown in Equation 3.8. As mentioned in Chapter 3, similarly to the gradient of the IWE, this loss aims to capture regions of high-intensity change, which should in theory recover the edges that generate the events, leading to the IWE.

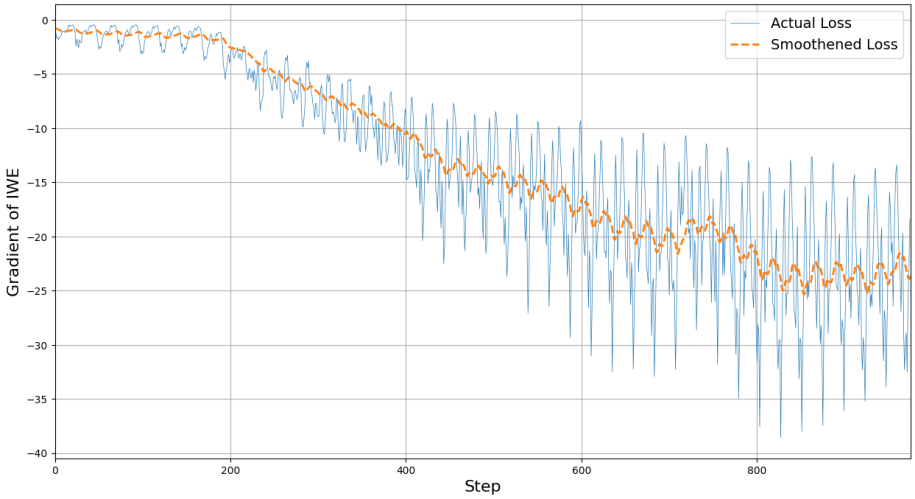


Figure 4.5: Loss progression during training, using the Laplacian of the IWE.

Despite undergoing similar trends, such as this initial region of no learning, the Laplacian of the IWE is able to converge faster compared to the previously tested loss function. Unfortunately, this also leads to the networks following the same trends when predicting optical flow as shown in Figure 4.6

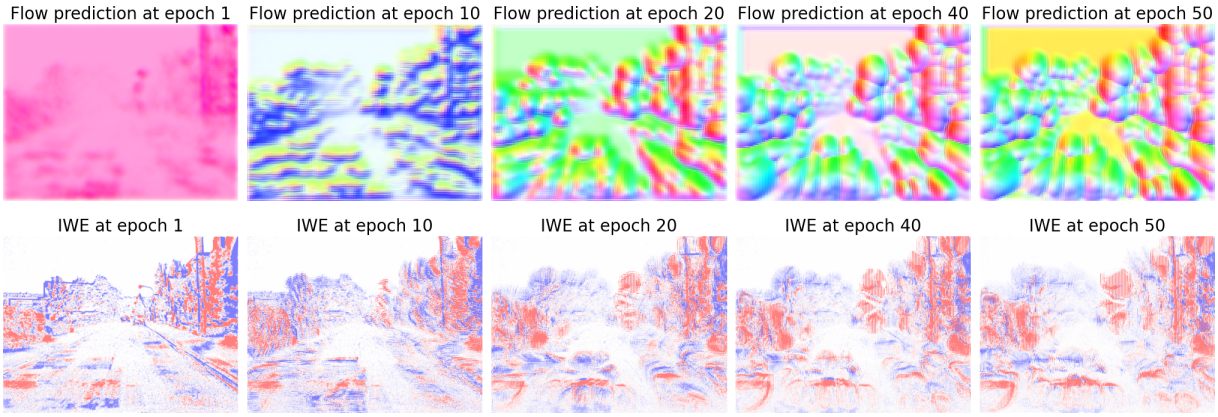


Figure 4.6: Optical flow prediction and corresponding IWE recorded during training when optimizing the Laplacian of the IWE.

The horizontal stripe patterns present in Figure 4.4 are missing here, but the non-physical flow prediction remains prominent.

In general, these results bring about an important finding regarding this self-supervised learning framework. Namely, the spatial contrast maximization model that is based on these statistical measures of the IWE tends to strongly overfit, transporting events to regions that maximize the contrast, but that do not abide by the physical constraints of the flow as already noted by [25]. However, during these experiments, the IWE was generated by deblurring the voxel grid that was produced by accumulating the event stream. In contrast to that, [25, 23] produce the IWE by first deblurring the raw events and then accumulating them to create the IWE. Therefore, additional research that utilizes that formulation is needed to assess the applicability of the spatial contrast maximization framework to IDNet fully.

4.2. Temporal Contrast Maximization

While the idea that maximizing the edge alignment of the IWE produces the best flow estimate is key here as well, the main observation that governs the operation of the temporal contrast maximization framework is that this is best achieved by minimizing the per-pixel, per-polarity average timestamp of the warped events, reflected in Equation 3.14.

Therefore, the loss formulation was implemented in IDNet's training framework using the same training settings described in Section 4.1 with the important difference that instead of using a fixed time frame of 100 [ms] to accumulate events, the samples were generated using a constant frame of 150000 events, which allows for efficient batching and GPU utilization. Training IDNet using that configuration until converging leads to the following loss progression, as seen in Figure 4.7

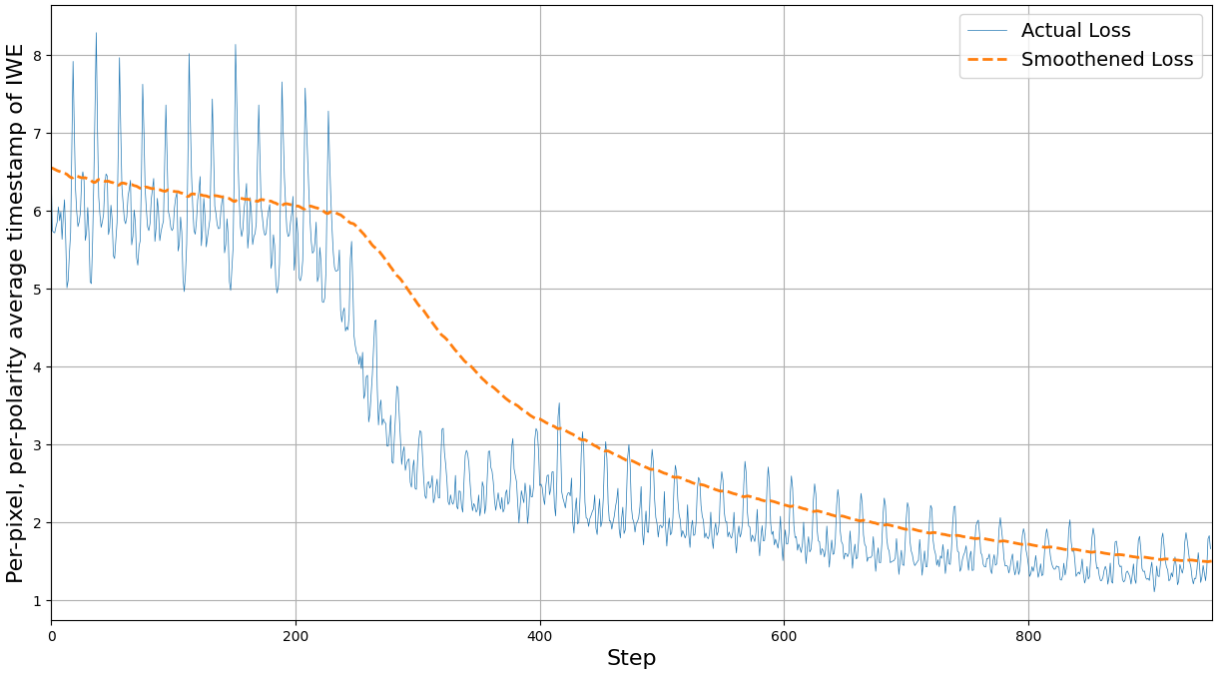


Figure 4.7: Loss progression during training, using the the loss described in Equation 3.14.

Again, the loss progression looks promising. Despite experiencing some difficulty learning during the first 200 steps, it quickly picks up speed and gradually plateaus around step 1000, indicating convergence. As shown before, the training loss is not the whole picture in the case of self-supervised learning, and therefore, inspecting the flow progression is of even greater importance. This progression can be seen in Figure 4.8.

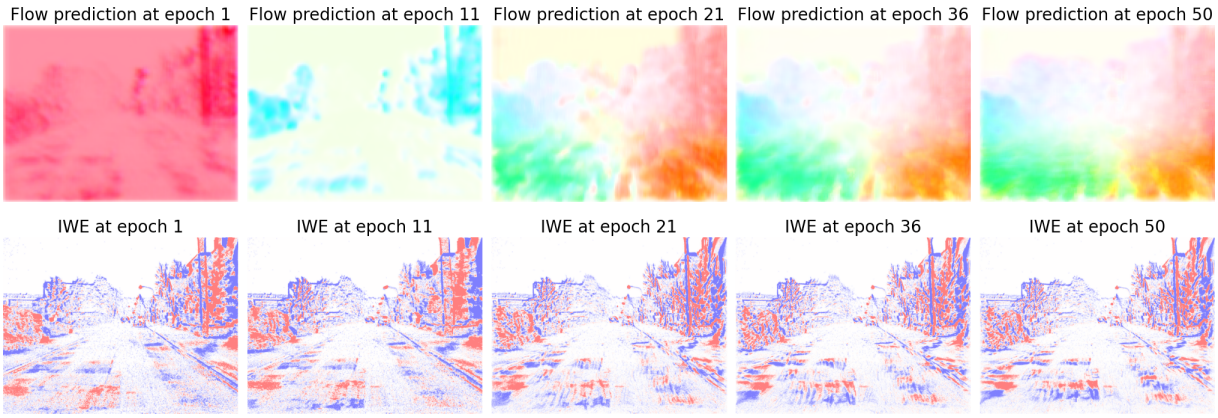


Figure 4.8: Optical flow prediction and corresponding IWE recorded during training when optimizing Equation 3.14.

While the first two columns representing epochs 1 and 11 look just as physically inconsistent as the previous training attempts, the flow prediction from epoch 21 looks surprisingly much more accurate, which only improves into epochs 36 and 50. Considering that the scenes present in $zurich\_city\_01\_a$ represent a car driving forward, the flow patterns and colors match what the expected results should be.

Despite the lack of detail in the flow map, which could be attributed to the fact that the network was only trained on a single sequence, the predictions seem to be consistent with the physical scenario, which leaves a promising trail to integrating the temporal contrast maximization loss fully into IDNet.

5

Conclusion

The research objective formulated at the start of this research work was to expand the capabilities of IDNet by removing its dependency on labeled data and reforming its refinement scheme. The process started with an investigation into the existing literature surrounding the problems of event-based optical flow, self-supervised learning, and adaptive inference mechanisms. Based on that investigation, the methodology was then developed. This was a step that required many iterations and tests to converge to a working solution.

Self-Supervised Learning

Regarding the problem of self-supervised learning, initially, spatial contrast maximization (SCM) methods were implemented. These methods included maximizing the variance of the Image of Warped Events (IWE), the gradient of the IWE, and the Laplacian of the IWE. All of these losses were found to lead to strong overfitting behavior, which produced non-physical flow, even after convergence. This flow was characterized by the formation of ”bubble”-like structures, which indeed led to higher contrast but were far from the correct flow. Therefore, that direction was abandoned and more focused was paid to temporal contrast maximization (TCM). Instead of computing spatial statistics such as the variance, TCM maximizes the edge alignment of the IWE (and thus its contrast) by minimizing the per-pixel, per-polarity average timestamp of the warped events. Initial experiments with this loss showed promising results, which led to its full integration into the training pipeline of IDNet. After training on MVSEC, we notice improved results over previous methods with up to 15% on some sequences and an 8% improvement on average.

Deep Equilibrium Models

To convert the fixed-iteration scheme of IDNet into an adaptive one, Deep Equilibrium Models (DEQ) was the concept that showed the most potential. The idea behind it is to reformulate the internal architecture of the network as an infinitely deep implicit layer whose outputs are the solutions of a fixed-point equation, and precisely, these fixed points are the desired ”equilibrium” flow. This also allows for the utilization of traditional root-finding methods such as Anderson’s method, which permits the network to adaptively change the number of iterations based on a tolerance criteria set, closely resembling traditional optimization-based techniques. Moreover, these solvers exhibit super-linear behavior, allowing for much faster convergence. Another big benefit of that formulation was the application of the Implicit Function Theorem (IFT), which allowed us to directly differentiate through the network using only the final fixed-point solution and without any knowledge of previous iterations, which provided a theoretical reduction in memory from $O(L)$ to $O(1)$, where L is the number of fixed iterations IDNet performs. We further alleviated the training costs by approximating the Jacobian inversion present in the IFT, which, in practice, granted us an almost free backward pass compared to the forward pass. To stabilize the convergence of this model, we also involved a fixed-points correction scheme, which sampled the trajectory of the solver and computed a loss term with respect to each of these samples, motivating early convergence. Warm-starting is another strategy we used to speed up the training process by reutilizing previous solutions in order to give a better initial guess for the fixed-point solver, dramatically decreasing the number of iterations needed on subsequent passes. The final idea we integrated was that of tolerance scheduling. As the network got better at estimating the flow, we noticed that the iterations needed abruptly dropped, which was a sign that the network was not utilizing its full resources. Therefore, as that happened, we progressively decreased the tolerance

of the solver to squeeze out extra performance. Having researched and implemented all of these ideas, we trained and tested our DEQ model on the DSEC dataset. We found that our model was competitive, placing fourth behind state-of-the-art models such as IDNet and E-RAFT while providing an improvement in memory consumption of 15%. We theorize that this non-optimal performance could be attributed to the gradient approximation and raise this as a direction for further research.

6

Recommendations

This chapter provides a brief overview of the primary recommendations for the future continuation of this research project. These recommendations are split into two directions, covering the problems of self-supervised learning and deep equilibrium models.

6.1. Self-Supervised Learning

Using larger datasets

Since, in this training paradigm, there is a complete absence of ground truth data, the learning process occurs much more slowly. Therefore, more and more data is needed to allow the network to learn the correct patterns. We recommend including more datasets such as DSEC [13] and BlinkFlow [14], which include diverse sequences from urban and country-side environments captured in both day and night conditions. This could allow the network to learn even more complex flow patterns, further improving its performance.

Improving the loss

Utilizing the contrast maximization framework described in this report was proven to be an excellent candidate for the task of learning optical flow without ground truth data. Still, there has been recent research aiming at further improving it. Paredes-Valles et al. [27] build further on that framework by introducing an iterative event warping module and a multi-timescale loss function to increase its robustness to noise and non-linearities in the data. Therefore, integrating that into the current self-supervised learning framework of IDNet could lead to even better results.

Spatial Contrast Maximization

Despite the strong overfitting behavior we experienced while utilizing spatial contrast maximization, theoretically, it is still a good candidate for the task of self-supervised learning. We propose further research into regularizing mechanisms that would prevent that overfitting from occurring and allow the network to truly learn optical flow patterns.

6.2. Deep Equilibrium Models

Gradient approximation

One of the potential reasons why our DEQ model did not reach state-of-the-art performance was attributed to the gradient computation procedure, which approximated the Jacobian inversion. This could have led to a substantial loss of information during training and prevented the network from utilizing its full potential. Therefore, researching other ways of approximating that term is a research direction that could greatly benefit the DEQ formulation.

Hyperparameter tuning

During our experiments with the DEQ formulation of IDNet, we found that it is quite sensitive to its hyperparameters. Particularly, the tolerance of the solvers proved to be a critical parameter that dictated the performance of the network. Therefore, further research and tuning of that variable could allow for extra performance gains.

Testing different datasets

Another possibility as to why the DEQ formulation could not achieve SOTA results could lie in the data itself. In order for the network to successfully learn, the architecture must be able to pick up enough information from data in order to facilitate training. To investigate whether this is indeed a problem in this case, further experiments have to be performed on different datasets. Potential candidates for that are again MVSEC, which is a standard benchmark dataset in event-based vision [12], as well as the more recent BlinkFlow, which provides a gigantic amount of labeled data [14].

References

[1] Junjie Huang et al. *Optical Flow Based Real-time Moving Object Detection in Unconstrained Scenes*. arXiv:1807.04890 [cs]. July 2018. URL: http://arxiv.org/abs/1807.04890 (visited on 11/05/2023).

[2] G. C. H. E. de Croon et al. "Enhancing optical-flow-based control by learning visual appearance cues for flying robots". en. In: *Nature Machine Intelligence* 3.1 (Jan. 2021). Number: 1 Publisher: Nature Publishing Group, pp. 33–41. DOI: 10.1038/s42256-020-00279-7. URL: https://www.nature.com/articles/s42256-020-00279-7 (visited on 10/25/2023).

[3] Simon Baker et al. "Lucas-Kanade 20 Years On: A Unifying Framework". en. In: *International Journal of Computer Vision* 56.3 (Feb. 2004), pp. 221–255. DOI: 10.1023/B:VISI.0000011205.11775.fd. URL: http://link.springer.com/10.1023/B:VISI.0000011205.11775.fd (visited on 10/29/2023).

[4] Berthold K. P. Horn et al. "Determining optical flow". In: *Artificial Intelligence* 17.1 (Aug. 1981), pp. 185–203. DOI: 10.1016/0004-3702(81)90024-2. URL: https://www.sciencedirect.com/science/article/pii/0004370281900242 (visited on 11/06/2023).

[5] Deqing Sun et al. *PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume*. arXiv:1709.02371 [cs]. June 2018. URL: http://arxiv.org/abs/1709.02371 (visited on 10/29/2023).

[6] Zachary Teed et al. *RAFT: Recurrent All-Pairs Field Transforms for Optical Flow*. arXiv:2003.12039 [cs]. Aug. 2020. URL: http://arxiv.org/abs/2003.12039 (visited on 10/29/2023).

[7] Guillermo Gallego et al. "Event-based Vision: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.1 (Jan. 2022). arXiv:1904.08405 [cs], pp. 154–180. DOI: 10.1109/TPAMI.2020.3008413. URL: http://arxiv.org/abs/1904.08405 (visited on 10/29/2023).

[8] Alex Zihao Zhu et al. "EV-FlowNet: Self-Supervised Optical Flow Estimation for Event-based Cameras". In: *Robotics: Science and Systems XIV*. arXiv:1802.06898 [cs]. June 2018. DOI: 10.15607/RSS.2018.XIV.062. URL: http://arxiv.org/abs/1802.06898 (visited on 12/03/2023).

[9] Mathias Gehrig et al. "E-RAFT: Dense Optical Flow from Event Cameras". en. In: *2021 International Conference on 3D Vision (3DV)*. London, United Kingdom: IEEE, Dec. 2021, pp. 197–206. DOI: 10.1109/3DV53792.2021.00030. URL: https://ieeexplore.ieee.org/document/9665931/ (visited on 11/06/2023).

[10] Yilun Wu et al. *Rethinking Event-based Optical Flow: Iterative Deblurring as an Alternative to Correlation Volumes*. arXiv:2211.13726 [cs]. Mar. 2023. URL: http://arxiv.org/abs/2211.13726 (visited on 11/06/2023).

[11] Elias Mueggler et al. "The Event-Camera Dataset and Simulator: Event-based Data for Pose Estimation, Visual Odometry, and SLAM". In: *The International Journal of Robotics Research* 36.2 (Feb. 2017). arXiv:1610.08336 [cs], pp. 142–149. DOI: 10.1177/0278364917691115. URL: http://arxiv.org/abs/1610.08336 (visited on 11/20/2023).

[12] Alex Zihao Zhu et al. "The Multivehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception". In: *IEEE Robotics and Automation Letters* 3.3 (July 2018). Conference Name: IEEE Robotics and Automation Letters, pp. 2032–2039. DOI: 10.1109/LRA.2018.2800793. URL: https://ieeexplore.ieee.org/document/8288670 (visited on 11/06/2023).

[13] Mathias Gehrig et al. "DSEC: A Stereo Event Camera Dataset for Driving Scenarios". en. In: *IEEE Robotics and Automation Letters* 6.3 (July 2021), pp. 4947–4954. DOI: 10.1109/LRA.2021.3068942. URL: https://ieeexplore.ieee.org/document/9387069/ (visited on 11/20/2023).

[14] Yijin Li et al. *BlinkFlow: A Dataset to Push the Limits of Event-based Optical Flow Estimation*. arXiv:2303.07716 [cs]. Mar. 2023. DOI: 10.48550/arXiv.2303.07716. URL: http://arxiv.org/abs/2303.07716 (visited on 11/20/2023).

[15] J. L. Barron et al. "Performance of optical flow techniques". en. In: *International Journal of Computer Vision* 12.1 (Feb. 1994), pp. 43–77. DOI: 10.1007/BF01420984. URL: https://doi.org/10.1007/BF01420984 (visited on 11/14/2023).

[16] Philipp Fischer et al. *FlowNet: Learning Optical Flow with Convolutional Networks*. arXiv:1504.06852 [cs]. May 2015. URL: http://arxiv.org/abs/1504.06852 (visited on 11/19/2023).

[17] Eddy Ilg et al. *FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks*. arXiv:1612.01925 [cs]. Dec. 2016. URL: http://arxiv.org/abs/1612.01925 (visited on 11/19/2023).

[18] Alex Zihao Zhu et al. *Unsupervised Event-based Learning of Optical Flow, Depth, and Egomotion*. arXiv:1812.08156 [cs]. Dec. 2018. URL: http://arxiv.org/abs/1812.08156 (visited on 11/06/2023).

[19] Haotian Liu et al. *TMA: Temporal Motion Aggregation for Event-based Optical Flow*. arXiv:2303.11629 [cs]. Aug. 2023. URL: http://arxiv.org/abs/2303.11629 (visited on 12/02/2023).

[20] Guillermo Gallego et al. "A Unifying Contrast Maximization Framework for Event Cameras, with Applications to Motion, Depth, and Optical Flow Estimation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. arXiv:1804.01306 [cs]. June 2018, pp. 3867–3876. DOI: 10.1109/CVPR.2018.00407. URL: http://arxiv.org/abs/1804.01306 (visited on 12/03/2023).

[21] Vitor Guizilini et al. *Learning Optical Flow, Depth, and Scene Flow without Real-World Labels*. arXiv:2203.15089 [cs]. June 2022. DOI: 10.48550/arXiv.2203.15089. URL: http://arxiv.org/abs/2203.15089 (visited on 12/04/2023).

[22] Pengpeng Liu et al. *SelFlow: Self-Supervised Learning of Optical Flow*. arXiv:1904.09117 [cs]. Apr. 2019. URL: http://arxiv.org/abs/1904.09117 (visited on 12/04/2023).

[23] Guillermo Gallego et al. "Focus Is All You Need: Loss Functions For Event-based Vision". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. arXiv:1904.07235 [cs]. June 2019, pp. 12272–12281. DOI: 10.1109/CVPR.2019.01256. URL: http://arxiv.org/abs/1904.07235 (visited on 12/04/2023).

[24] Timo Stoffregen et al. "Event Cameras, Contrast Maximization and Reward Functions: An Analysis". en. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, June 2019, pp. 12292–12300. DOI: 10.1109/CVPR.2019.01258. URL: https://ieeexplore.ieee.org/document/8954356/ (visited on 12/04/2023).

[25] Shintaro Shiba et al. *Secrets of Event-Based Optical Flow*. arXiv:2207.10022 [cs]. July 2022. URL: http://arxiv.org/abs/2207.10022 (visited on 12/04/2023).

[26] Jesse Hagenaars et al. "Self-Supervised Learning of Event-Based Optical Flow with Spiking Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 7167–7179. URL: https://proceedings.neurips.cc/paper_files/paper/2021/hash/39d4b545fb02556829aab1db805021c3-Abstract.html (visited on 12/05/2023).

[27] Federico Paredes-Vallés et al. *Taming Contrast Maximization for Learning Sequential, Low-latency, Event-based Optical Flow*. arXiv:2303.05214 [cs]. Sept. 2023. URL: http://arxiv.org/abs/2303.05214 (visited on 12/05/2023).

[28] Yi Tian. "Event Transformer FlowNet for optical flow estimation". en. In: ().

[29] Shaojie Bai et al. *Deep Equilibrium Models*. arXiv:1909.01377 [cs, stat]. Oct. 2019. URL: http://arxiv.org/abs/1909.01377 (visited on 12/05/2023).

[30] Shaojie Bai et al. *Deep Equilibrium Optical Flow Estimation*. arXiv:2204.08442 [cs]. Apr. 2022. URL: http://arxiv.org/abs/2204.08442 (visited on 12/05/2023).

[31] Bac Nguyen et al. *Efficient Training of Deep Equilibrium Models*. arXiv:2304.11663 [cs]. Apr. 2023. URL: http://arxiv.org/abs/2304.11663 (visited on 12/06/2023).

A

Appendix

Optical Flow Coding Scheme

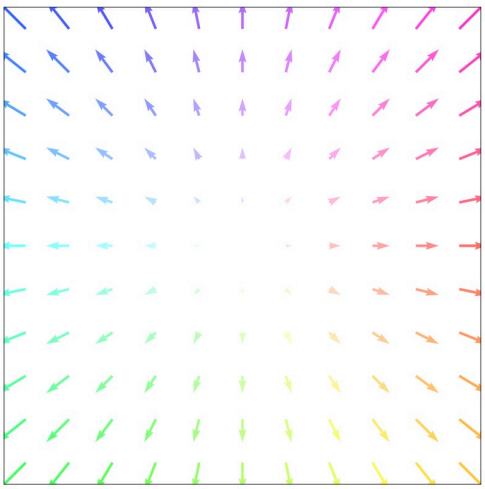


Figure A.1: Optical flow coding scheme where the magnitude of the optical flow is encoded in brightness and the direction in color hue.