A Comparative Study of Model-based and Learning-based Optical Flow Estimation methods with Event Cameras

David Dinucu-Jianu[1]

Supervisor(s): Nergis Tömen[1]**, Hesam Araghi**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2024

Name of the student: David Dinucu-Jianu
Final project course: CSE3000 Research Project
Thesis committee: Nergis Tömen, Hesam Araghi, Guohao Lan

Abstract

Optical flow estimation with event cameras encompasses two primary algorithm classes: model-based and learning-based methods. Model-based approaches, do not require any training data while learning-based approaches utilize datasets of events to train neural networks. To effectively apply these algorithms, it's essential to understand their respective strengths and weaknesses. This study compares model-based and learning-based optical flow estimation methods using event cameras, aiming to provide guidance for real-world applications. We evaluated these methods on the MVSEC and DSEC datasets, focusing on their accuracy and runtime. Our findings indicate that model-based methods excel on the MVSEC dataset, characterized by small motions, while learning-based approaches perform better on the more dynamic DSEC dataset. To investigate potential overfitting of learning-based methods to DSEC, we retrained the IDNet and TMA models on the BlinkFlow dataset. The retrained models demonstrated competitive accuracy, surpassing model-based methods which indicates that learning-based models perform better on datasets like DSEC even when not able to overfit. Finally, our analysis on runtime showed that model-based methods achieve real-time performance on CPUs and learning-based methods require a GPU to run in real-time.

1 Introduction

Optical flow estimation is a fundamental computer vision task [1]. It can aid in compression algorithms [2], in object tracking [3], video restoration [4] and in robotics where optical flow helps in the detection and avoidance of obstacles [5, 6]. Event cameras are particularly suitable for optical flow estimation due to their detailed temporal information. Unlike standard cameras, event cameras produce a stream of asynchronous per-pixel brightness changes, known as "events," which encode the position, time, and polarity of the brightness change. This high temporal resolution [7] allows for capturing rapid motion and fine-grained details, making event cameras ideal for accurate and efficient optical flow estimation.

In the context of event cameras, optical flow needs to be predicted from the sequence of events instead of frames leading to significant challenges in the adaptation of methods from frame based cameras to the new medium.

There exist two main families of algorithms for event-based optical flow:

1. **Model-based Algorithms**: These algorithms investigate the characteristics of event data and formulate mathematical models to compute the flow.
2. **Learning-based Approaches**: These approaches utilize datasets of events to train neural networks to predict the flow. However, obtaining real-world ground truth data for optical flow is challenging due to the complexity of accurately capturing and annotating precise motion information in dynamic scenes. Consequently, if the accuracy of model-based methods is sufficient, the use of model-based methods is often preferred.

Previously, learning-based approaches were widely regarded as more accurate than model-based methods. However, this assumption has been challenged by the introduction of MultiCM [8], which outperforms all other methods on the MVSEC dataset [9]. Despite this success, MultiCM, along with another leading model-based method by Brebion et al. [10], significantly underperformed on the DSEC dataset [11]. This discrepancy raises important questions about the conditions under which one approach may be better than the other and how they compare in terms of accuracy, runtime, and generalizability.

Shiba et al. [12] theorized that the accuracy gap between the MVSEC and DSEC datasets for model-based methods compared to learning-based approaches might be due to overfitting. Specifically, learning-based methods may overfit to the forward motion prevalent in the DSEC dataset, which primarily contains driving scenarios, thereby inflating their performance scores. In order to explore this theory, we will train two models, IDNet [13] and TMA [14], on BlinkFlow [15], which is another dataset that does not have this forward motion bias. By testing these models on DSEC, we can measure the magnitude of this effect and determine if forward motion is the only factor that leads to the better performance of learning-based methods or if there are other reasons for this gap.

By assessing the accuracy and runtime of these methods and examining the performance discrepancies between the MVSEC and DSEC datasets for model-based algorithms, we aim to offer practical guidance for real-world applications. This exploration will help identify the factors contributing to the performance gaps and highlight areas for potential improvements in current algorithms.

This research aims to compare model-based and learning-based approaches for optical flow estimation with event cameras and explore the performance gap observed on the MVSEC and DSEC datasets by addressing the following questions:

1. How do model-based and learning-based approaches compare in terms of accuracy on publicly available datasets?

2. How do the two approaches compare in terms of runtime performance?

3. Why do model-based approaches excel on MVSEC but perform substantially worse on DSEC when compared to learning-based methods?

2 Related Work

Optical flow estimation methods for event cameras can be categorized into two primary approaches: model-based and learning-based. Each approach has distinct characteristics and methodologies.

2.1 Model-based approaches

Model-based approaches do not need any training data and rather rely on the underlying physical and geometric principles of motion. These methods focus on mathematical modeling of event sequences, often using algorithms to optimize objective functions that describe the motion of detected events over time. These approaches generally rely on assumptions to model event motion, the most common being Brightness Constancy, which assumes that the brightness of a pixel remains consistent throughout its motion and the local constant flow assumption which asserts that the flow is constant over a neighbourhood around a pixel. However, such assumptions may not always hold, potentially leading to underperformance.

Types of model-based algorithms

There is a significant variety within model-based approaches for event-based optical flow estimation. Early methods adapted traditional frame-based algorithms to handle event data, while more recent methods leverage the unique characteristics of event data to achieve greater accuracy and efficiency. Notable examples of these approaches include:

- **Lucas-Kanade Extension:** Benosman et al. [16] extended the classic Lucas-Kanade method [17] to event based optical flow. This method involves predicting optical flow for each event by defining a neighborhood of $n \times n \times \Delta t$, where n is typically 5 and Δt is around $60\mu s$. Within this neighborhood, spatial and temporal derivatives of event activities are computed. These derivatives form a system of equations for each event, which can be solved using least squares minimization to estimate the optical flow components. This algorithm is simple, can be efficiently implemented on CPUs and GPUs but bases itself on the Brightness Constancy and the constant local flow assumption.
- **Block-Matching Techniques:** Liu et al. [18,19] adapted block-matching techniques for use with event cameras. These methods involves dividing the sensor's output into smaller blocks and tracking the movement of these blocks over time. Events are accumulated into time slices, and the motion of each block is determined by comparing the current block of events to potential matches within a search area in a previous time slice. This method has been adapted to work successfully on FPGAs allowing it to have great power efficiency and latency, furthermore as opposed to Lucas-Kanade methods [16], the method can account for non-local structures in the video.
- **Plane Fitting Methods:** Plane fitting methods [20] estimate visual motion flow by fitting a plane to the spatio-temporal data of events within a local neighborhood. These methods assume that within a small spatial and temporal window, the velocity is constant, leading to a locally planar representation of the event surface.

  For each incoming event, a spatio-temporal window is defined around the event. A least squares minimization is applied to fit a plane to the events in this window. The plane parameters represent the trajectory of the events.

  Additionally, these methods can use multi-scale pooling techniques [21] in order to improve accuracy. This involves computing local flow estimates at multiple spatial scales and selecting the scale with the most consistent flow magnitude.

  These approaches are sensitive to noise and run for each incoming event which could cause problems if the events are very dense. They are however, efficient, easy to implement and robust against the aperture problem.
- **Contrast Maximization Framework:** Contrast maximization (CM) is a technique that estimates motion by optimizing the sharpness of motion-compensated images. In CM, events are warped along predicted point trajectories to create an image of warped events (IWE). The objective function evaluates the contrast of this IWE, with higher contrast indicating better alignment of events or pixels and thus more accurate motion estimation. The optimization seeks to find the warp parameters that maximize this contrast.

  However, applying CM to optical flow estimation faces challenges such as overfitting and event collapse. The framework can lead to overfitting by pushing events into a line, particularly in scenes dominated by line features, resulting in what is known as the aperture problem. This occurs because line segments moving across the image plane can produce ambiguous optical flow estimates, with multiple trajectories producing similar contrast maximization results.

  MultiCM [8, 12] tackles these issues by introducing a novel loss function to enhance accuracy and reduce overfitting, improving handling of occlusions, and adopting a multiscale approach to boost convergence and prevent the algorithm from getting trapped in local optima. Despite being a state-of-the-art method on the MVSEC [9] dataset, its current runtime performance is not desirable, which may result from an unoptimized implementation rather than inherent complexity [8]. MultiCM is the leading algorithm in terms of accuracy for model-based methods and will be a central focus of this research.
- **Pipeline Method with Inverse Exponential Distance Surface:** Brebion et al. [10] developed a versatile framework for real-time optical flow computation with both low- and high-resolution event cameras. This method begins by accumulating events into edge images over short temporal windows using the CPU. These edge images are then denoised to remove isolated noise and filled to stabilize the representation by adding missing edge pixels where neighboring pixels indicate an edge should be present. The "inverse exponential distance surface" converts these edge images into a dense format suitable for frame-based optical flow algorithms, re-

ducing noise impact and preserving object edges. The current optical flow computation in the pipeline utilizes a predictive filter-based approach [22], which can operate on either the CPU or be optimized for GPU implementation, achieving real-time performance with frame rates of 250Hz at 346×260 pixels and 77Hz at 1280×720 pixels. The modular nature of this method allows for substituting the optical flow algorithm at the end of the pipeline with other methods as needed. This method will be further explored in our study due to its impressive runtime performance and strong accuracy on benchmarks, falling just behind MultiCM.

2.2 Learning-based approaches

Learning-based approaches are extensively utilized in optical flow estimation for both frame-based and event-based cameras. These methods offer several advantages over model-based approaches, including higher accuracy, competitive runtime, and independence from assumptions like Brightness Constancy. However, these benefits come at the cost of requiring substantial training data, which is often expensive and may lack reliable ground truth.

Input Representation

To process event data effectively, algorithms can adopt one of two strategies. One approach is utilizing Spiking Neural Networks (SNNs), networks that mimic biological neurons and can process event data directly without preprocessing. Models such as [23] are based on this strategy. Alternatively, the data can be transformed into a structured format, such as voxel grids, making it compatible with traditional neural network architectures. Voxel grids are the most popular representation in the current landscape of event-based algorithms and involve computing a 3D grid where each voxel accumulates the number or properties of events that occur within its spatial and temporal boundaries. Figure 1 provides a visualization of how a voxel grid representation looks like for the DSEC dataset.

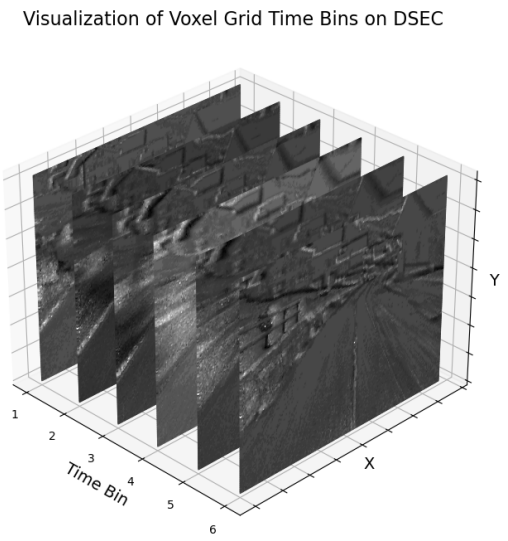

Figure 1: Illustration of a voxel grid representation. The 2D spatial dimensions correspond to the resolution of the event camera sensor, and the third dimension represents time divided into bins. Each voxel contains aggregated information about the events that occurred within its volume.

Categories of Learning Approaches

Learning-based optical flow estimation algorithms are categorized into three main types based on the type of learning they perform:

- **Supervised Learning:** These methods [13–15, 24] rely on labeled ground truth data to train neural networks for accurate optical flow prediction. While effective, they are limited by the availability and quality of ground truth data, which can vary across datasets.
- **Self-Supervised Learning:** In contrast to supervised learning, self-supervised approaches [25–27] only utilize event data to learn to predict the optical flow. This eliminates the need for labeled data, making them more adaptable to diverse environments. However, achieving high accuracy without ground truth supervision remains a challenge.
- **Semi-Supervised Learning:** Semi-supervised methods [28] combine the strengths of supervised and self-supervised approaches by incorporating auxiliary signals, such as grayscale images, alongside event data during training. This hybrid approach can improve accuracy by leveraging additional information while mitigating the need for extensive labeled data.

For the purposes of this study, we will focus primarily on supervised learning and self-supervised learning. Semi-supervised learning, while beneficial, requires additional data sources such as grayscale images, which may not always be available or applicable in our targeted environments. By concentrating on supervised and self-supervised methods, we aim to address the broader applicability of optical flow estimation using event data alone.

Notable Learning-Based Methods in Optical Flow Estimation

Initially, optical flow methods for event-based cameras adapted architectures from frame-based applications and used self-supervised and semi-supervised learning. For example, the U-Net [29] architecture was utilized for event data by Zhu et al. [25] and the EV-FlowNet model [28]. EV-FlowNet [28] is a particularly interesting model that used semi-supervised learning by using the grayscale images provided from the MVSEC dataset [9] and optimized the model using a photometric and smoothness loss.

The introduction of the DSEC dataset [11] made supervised learning more feasible by providing higher resolution and more reliable ground truth flow, thus a significant increase in terms of accuracy was marked by the adaptation of the RAFT [30] architecture into the E-RAFT model [24], originally developed for frame-based optical flow. E-RAFT was one of the first methods to use correlation volumes in the event camera literature and represented events as Voxel Grids.

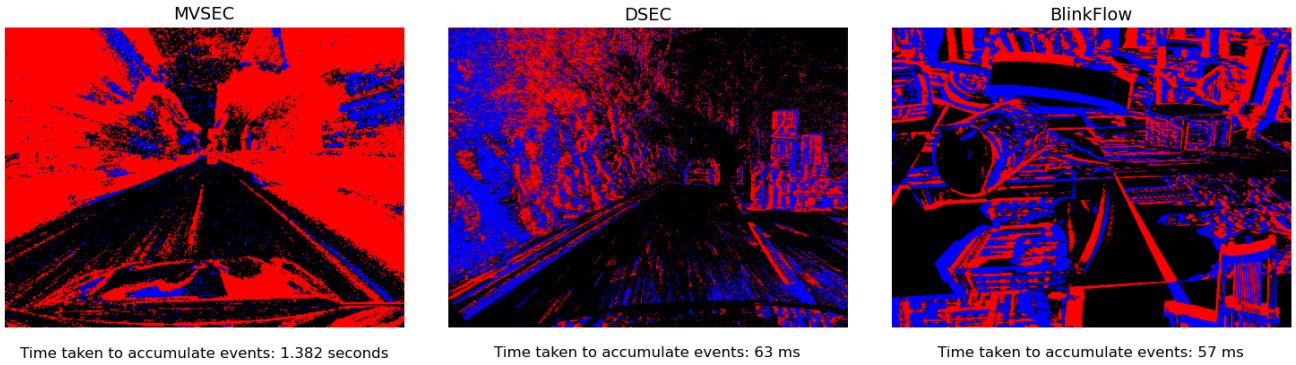

Figure 2: Illustration of one million accumulated events from each of the datasets. Blue represents a predominantly positive negative polarity for the pixel, and red represents a positive polarity. It can be seen that MVSEC takes two orders of magnitude more time for one million events to accumulate than DSEC and BlinkFlow.

Subsequent research led to models such as E-FlowFormer by Li et al. [15], which employs transformers [31] to enhance feature encoding and is currently one of the leading models in terms of accuracy. In addition to this, Temporal Motion Aggregation (TMA) [14] leverages the temporal continuity inherent in event data to significantly increase accuracy, thus positioning it alongside E-FlowFormer [15] as a state-of-the-art model.

Furthermore, the Iterative Deblurring Network (IDNet) [13] represents an efficient approach that differentiates itself from other methods by avoiding the use of correlation volumes which are especially costly to compute. This design choice results in a lightweight model with competitive accuracy, demonstrating an alternative path for optimizing performance and efficiency in event-based optical flow estimation. This method will be further explored in our study for its efficiency and great accuracy.

Another notable method is Taming Contrast Maximization (TamCM) [26], which combines ideas from traditional contrast maximization with modern learning-based techniques to perform self-supervised learning efficiently. This approach leverages the strengths of contrast maximization to align events accurately while using learning-based methods to refine the optical flow estimation and will be our baseline for the self-supervised category of models.

3 Performance & Runtime Evaluation

To compare the accuracy and runtime performance of learning-based and model-based approaches, we will evaluate them on publicly available datasets. In Section 3.1, we will introduce the datasets commonly used in the literature. Subsequently, we will present the evaluation results, focusing on accuracy in Section 3.2 and inference speed in Section 3.3.

3.1 Datasets

In the event-based optical flow literature, three datasets are commonly used: MVSEC [9], DSEC [11], and the newly released BlinkFlow [15]. This study compares the performance of different methods on the MVSEC and DSEC datasets, aggregating results where reported. BlinkFlow was excluded from the study due to the lack of ground truth in public test data and the scarcity of benchmarks for other methods on this dataset. Additionally, the training data for BlinkFlow is not very diverse and is significantly different from the test data, which could make it more difficult to reliably assess performance. Figure 2 visualizes how events from the three datasets look like and how dense they are in time.

MVSEC

The Multi-Vehicle Stereo Event Camera (MVSEC) dataset is the first large-scale event-based optical flow dataset. This dataset posses several limitations. Gehrig et al. [24] highlighted that the magnitude of changes was tiny (less than 3 pixels) in the majority of sequences. Additionally, Shiba et al. [8] pointed out issues with the ground truth data due to the differing frequencies of the cameras and sensors involved in the creation of the groundtruth flow. The dataset includes sequences involving drones flying indoors and outdoors, driving scenarios during daytime and nighttime, and one sequence involving handheld motion.

DSEC

DSEC offers a large amount of high-quality training data for driving scenarios. It was created to address the limitations of the MVSEC dataset and has become a benchmark standard. DSEC adopted the use of a higher resolution sensor with a megapixel count of 1.6, in contrast to 0.1MP of the MVSEC dataset, it featured larger displacements and improved the ground truth data generation. The dataset consists entirely of driving scenarios captured during daytime and nighttime. Shiba et al. [8] noted that learning methods could overfit to the nature of the training data, as driving scenarios predominantly involve forward motion.

Table 1: Comparison of different methods on the MVSEC dataset for the scenarios: $indoor\_flying1$, $indoor\_flying2$,$indoor\_flying3$, and $outdoor\_day1$. Metrics measured include endpoint error (EPE) and the percentage of pixels with EPE greater than 3 pixels, with optical flow computed once every 22ms. All results displayed are reported from the respective papers; missing entries represent sections where the authors have not tested their model, bold entries represent the best score, while underlined ones are the second best.

		$indoor\_flying1$		$indoor\_flying2$		$indoor\_flying3$		$outdoor\_day1$	
		EPE ↓	$\%_{3PE}$ ↓	EPE ↓	$\%_{3PE}$ ↓	EPE ↓	$\%_{3PE}$ ↓	EPE ↓	$\%_{3PE}$ ↓
SL	E-RAFT [24]	1.10	5.72	1.94	30.79	1.66	25.20	**0.24**	**0.00**
	TMA [14]	1.06	3.63	1.81	27.29	1.58	23.26	<u>0.25</u>	0.07
	IDNet [13] (4 iterations,1/4 resolution)	-	-	-	-	-	-	0.31	0.1
	IDNet [13] (4 iterations,1/8 resolution)	-	-	-	-	-	-	0.34	**0.0**
	IDNet [13] (TID,1 iteration,1/8 resolution)	-	-	-	-	-	-	0.45	0.2
SSL	TamCM [26]	<u>0.44</u>	**0.00**	<u>0.88</u>	4.51	<u>0.70</u>	2.41	0.27	0.05
M	Nagata et al. [32]	0.62	-	0.93	-	0.84	-	0.77	-
	Brebion et al. [10]	0.52	<u>0.10</u>	0.98	<u>5.50</u>	0.71	<u>2.10</u>	0.53	0.20
	MultiCM [8]	**0.42**	<u>0.10</u>	**0.60**	**0.59**	**0.50**	**0.28**	0.30	0.10

SL - Models using supervised learning, **SSL** - Self-Supervised models, **M** - Model-based methods.

Table 2: Comparison of different methods on the DSEC dataset averaged over all test scenarios. Metrics measured are endpoint error (EPE) and the percentages of pixels with errors larger than 1, 2 and 3 pixels. The optical flow is computed at each 100ms. All results displayed are reported from the respective papers, bold entries represent the best score while underlined ones are the second best.

		EPE ↓	$\%_{1PE}$ ↓	$\%_{2PE}$ ↓	$\%_{3PE}$ ↓
Supervised	E-RAFT	0.788	12.742	4.74	2.684
	TMA	<u>0.743</u>	<u>10.863</u>	<u>3.972</u>	2.301
	IDNet (4 iterations, 1/4 resolution)	**0.719**	**10.069**	**3.497**	**2.036**
	IDNet (4 iterations, 1/8 resolution)	0.770	12.100	4.000	<u>2.200</u>
	IDNet (TID, 1 iteration, 1/8 resolution)	0.840	14.700	5.000	2.800
Self-Supervised	TamCM	2.330	68.293	33.481	17.771
Model-based	MultiCM	3.472	76.57	48.48	30.855
	Brebion et al.	4.881	82.812	57.901	41.952

BlinkFlow

BlinkFlow is the latest dataset, featuring fully simulated training data in an environment called BlinkSim. It provides the most reliable ground truth data, and the E-FlowFormer algorithm has demonstrated that, unlike previous works on simulated datasets, algorithms trained on BlinkFlow generalize well beyond the dataset. BlinkFlow includes a wide variety of motion types, not just forward motion like DSEC, but also more varied movements.

3.2 Accuracy assessment

We will assess the performance of the different methods by using the following metrics: the Endpoint Error (EPE) and the percentage of pixels with EPE larger than 1%, 2%, and 3% where EPE is defined as the average L2-norm of the optical flow error [33]. In this section, we will focus on the state-of-the-art models from model-based and learning-based methods but in Appendix A we also cover more classical approaches, namely Plane Fitting [20], Triplet Matching [34] and Time Gradient [35].

MVSEC Results

Model-based approaches outperform others on the MVSEC dataset in most scenarios and metrics. As shown in Table 1, model-based methods, especially MultiCM [8], consistently achieve lower endpoint errors (EPE) and fewer pixels with EPE greater than 3 compared to learning-based approaches. In particular, learning-based methods perform slightly better in the outdoor day1 scenario, possibly due to the violation of the Brightness Constancy assumption in variable outdoor lighting conditions.

DSEC Results

The results on DSEC reveal a significant gap between the leading model-based methods and supervised learning approaches. Shiba et al. [12] theorize that due to the nature of the DSEC dataset, which includes only driving scenarios, learning-based approaches overfit the predominant forward movement, resulting in high scores. Alternatively, it could be that model-based methods do not perform well on large pixel distances, which are much more prevalent in DSEC than in MVSEC.

It is important to note that the self-supervised method Taming Contrast Maximization (TamCM) [26] achieved higher accuracy than MultiCM. This could be because the losses used in TamCM help the model generalize better than the optimization procedures of model-based approaches. Additionally, by being trained on DSEC, TamCM may have learned that predicting forward motion generally yields higher scores, leading to a tendency to favor forward motion predictions without overfitting to the ground truth flow.

3.3 Runtime performance

The runtime performance of optical flow computation is crucial for the successful application of these approaches. In scenarios such as real-time processing on small, embedded hardware, such as drones, inference time is essential.

Table 3 presents the inference time for generating one flow from the DSEC dataset, which accumulates events over a 100-millisecond window. To simulate performance on mobile hardware, we conducted our tests using a laptop equipped with an AMD Ryzen 7 5800HS CPU and an RTX 3060 Laptop GPU.

Table 3: Inference time Comparison of different methods on DSEC Dataset. All benchmarks are performed on a laptop with an AMD Ryzen 7 5800HS CPU and an RTX 3060 Laptop GPU. Runtimes are averaged over 100 computations of flows.

		CPU	GPU
SL	E-RAFT (12 iterations, 1/8 resolution)	2.52s	130ms
SL	TMA	8.66s	246ms
SL	IDNet (4 iterations, 1/4 resolution)	7.70s	325ms
SL	IDNet (4 iterations, 1/8 resolution)	2.23s	120ms
SL	IDNet (TID, 1 iteration, 1/8 resolution)	530ms	24ms
M	MultiCM	>10s	>10s
M	Brebion et al.	63ms	39ms

SL - Models using supervised learning, **M** - Model-based methods.

It can be seen that the fastest method available is the model-based method created by Brebion et al. [10] which can run in realtime even on a CPU. The MultiCM [8] algorithm has a slow runtime which might be due to the unoptimized implementation of the current method but improvements could possibly be devised to make it run faster. On the Supervised learning side we can see that IDNet is able to run in realtime on GPUs which makes it a good option for applications that might need good accuracy and fast runtime.

The results indicate that, although learning-based methods may not be the absolute fastest, they offer sufficient efficiency for a wide range of applications while still achieving state-of-the-art performance.

4 Exploring the gap between MVSEC and DSEC

Since there exists a large performance gap for model-based methods between the MVSEC and DSEC datasets as exemplified by Table 1, 2, we aim to explore and reason about the existence of the gap.

There are two leading theories in the literature. On one hand, as Shiba et al. [8] theorizes the supervised learning methods might overfit to the nature of the DSEC dataset which includes only driving scenarios and the supervised methods might overfit to the forward motion. On the other hand, model-based methods might not be a good fit for datasets with large displacements such as DSEC and their limitations or assumptions might be limiting their performance instead. If the overfitting is the not the only factor we should expect to see the accuracy of the retrained models to still be greater than the model-based methods.

4.1 Experiment Setting

We aim to explore this by training a model from scratch on BlinkFlow, which does not only have forward motion but much more diverse motion types.

We decided to retrain IDNet [13] on the BlinkFlow dataset. We trained two versions, one at 1/4 and another at 1/8 resolution, for 25 epochs on the first three sets of sequences from the dataset, totaling a third of the entire dataset. We only used a subset of the complete dataset due to computational reasons. All training was done on a single NVIDIA RTX 3090 24GB RAM GPU and took lees than 12 hours. We used an Adam optimizer [36], with a learning rate of 1e-4 and a One Cycle scheduler. Almost all hyperparameters were adapted from the original IDNet paper [13].

Additionally, we trained the TMA [14] model for 25k steps(around 10 epochs), using AdamW [37] with a 2e-4 learning rate on an NVIDIA RTX A6000 GPU, also with a One Cycle scheduler and it took around 16 hours. Training the TMA [14] model aimed to determine whether the results are dependent on the specific model or if learning-based approaches in general achieve these results. Both models were trained entirely from scratch and do not have any pretrained feature encoders or modules.

4.2 Results

Table 4 demonstrates that both the retrained IDNet and TMA outperform the MultiCM algorithm, even though they were not trained on DSEC and the forward motion bias not being able to affect the results. Furthermore, the retrained algorithms achieves better accuracy than self-supervised methods trained on DSEC.

Despite the promising results, it is evident that the retrained IDNet and TMA algorithms still lag significantly behind the original models trained directly on DSEC. This suggests some level of overfitting to the specific characteristics of the DSEC dataset, possibly due to the motion type or

inherent differences between the BlinkSim simulator and the real-world event camera used in DSEC event generation.

Table 4: Comparison of methods on the DSEC dataset. The first three rows show results for the original IDNet and TMA trained on DSEC, followed by IDNet and TMA retrained on BlinkFlow. Taming Contrast Maximization (self-supervised) and MultiCM (model-based) are included for comparison.

		EPE ↓	$\%_{1PE}$ ↓	$\%_{2PE}$ ↓	$\%_{3PE}$ ↓
DSEC	IDNet (1/8)	0.770	12.100	4.000	2.200
	IDNet (1/4)	0.719	10.069	3.497	2.036
	TMA	0.743	10.863	3.972	2.301
BF	IDNet (1/8)*	1.964	58.522	27.664	14.139
	IDNet (1/4)*	1.844	47.657	22.657	12.594
	TMA*	1.938	51.618	21.111	9.693
SSL	TamCM	2.330	68.293	33.481	17.771
M	MultiCM	3.472	76.57	48.48	30.855

Models marked with an asterisk (*) were retrained on BlinkFlow. **BF** - Models trained on BlinkFlow, **DSEC** - Supervised models trained on DSEC, **SSL** - Models that are Self-Supervised, **M** - Model-based methods.

Considering these results and our initial hypothesis, we can assert that the superior accuracy of learning-based approaches, compared to model-based methods on the DSEC dataset, is not solely due to overfitting to forward motion. Instead, model-based approaches inherently perform worse on datasets like DSEC that feature large displacements. This performance gap indicates that learning-based methods are more adaptable and capable of handling complex motion dynamics present in such datasets.

5 Conclusion

This work provides a comprehensive comparison of model-based and learning-based optical flow estimation methods using event cameras, with a focus on their performance across the MVSEC and DSEC datasets. Model-based methods are shown to be more effective on the MVSEC dataset, which is characterized by smaller motions and shorter time intervals. In contrast, learning-based methods exhibit superior performance on the DSEC dataset, due to its dynamic motion patterns.

Our analysis suggests that the performance differences between model-based methods on the MVSEC and DSEC datasets stem from their inherent limitations, such as reliance on assumptions. We have shown this by retraining the IDNet and TMA models on the synthetic BlinkFlow dataset. Even though these retrained models did not achieve the same performance as those trained directly on DSEC, they still significantly outperformed model-based methods. This indicates that overfitting to the DSEC dataset does not fully explain the performance gap. Instead, it highlights that learning-based methods are inherently more adaptable and effective in handling complex motion dynamics.

While model-based methods such as those developed by Brebion et al. [10] can achieve real-time performance on CPUs, learning-based methods, although not optimized for CPU performance, still deliver decent results. However, their performance excels on GPU-equipped systems, indicating that with the right hardware, learning-based methods are well-suited for real-time applications.

5.1 Recommendations

Based on the findings of this study, we recommend the following for selecting optical flow estimation methods:

1. **Scenario-Based Method Selection:** For applications with small and predictable motions, model-based methods are preferable because of their efficiency and CPU-based real-time performance. These methods are ideal in environments with limited computing resources or where fast processing is essential.
2. **Use of Learning-Based Methods in Dynamic Environments:** In scenarios with complex and varied motions, such as dynamic driving or robotic navigation, learning-based methods are better suited due to their robustness and ability to generalize from diverse training data.
3. **Leveraging Synthetic Data for Training:** The success of models retrained on the BlinkFlow dataset supports the use of synthetic data to train learning-based optical flow models. This strategy reduces the need for extensive real-life data collection and can help models generalize more effectively across different settings.
4. **Exploring Dedicated Hardware:** Future research might explore the deployment of these methods on dedicated hardware like FPGAs and examine the runtime of learning-based approaches on such hardware.

These recommendations are intended to assist in the selection and application of optical flow estimation methods tailored to the specific needs and constraints of different scenarios.

6 Responsible Research

In this section, we address the reproducibility of our work, ensuring that our research adheres to high standards of integrity and transparency.

6.1 Reproducibility

To promote transparency and facilitate the validation of our findings, all experiments conducted in this study have been thoroughly documented and are fully reproducible. We have reported the hardware used to benchmark the methods, ensuring that other researchers can replicate our setup and results. The metrics used in our evaluation are standard for the field, providing a consistent basis for comparison. In addition, the evaluation of our new methods was performed using the DSEC website, where one can upload their predictions

directly[1]. This approach minimizes the possibility of code-related issues leading to incorrect results, as the evaluation platform ensures standardized and accurate assessments. We have also open-sourced[2] our retrained models on the BlinkFlow dataset, ensuring that the results can be verified and reproduced.

6.2 Ethical Considerations

Our study has been designed with careful attention to ethical considerations. We believe that our work does not pose significant ethical issues within the scope of this research. However, it is important to recognize that the broader field of optical flow and computer vision, particularly with the use of event cameras, can have ethical implications. Event cameras, which capture asynchronous visual data, can significantly enhance the performance of autonomous systems. This technology can be used for both beneficial and harmful purposes. Therefore, it is essential to develop such systems responsibly, ensuring that ethical guidelines are in place to govern their use.

References

[1] D. Fortun, P. Bouthemy, and C. Kervrann, "Optical flow modeling and computation: A survey," *Computer Vision and Image Understanding*, vol. 134, 02 2015.

[2] M. Jakubowski and G. Pastuszak, "Block-based motion estimation algorithms - a survey," *Opto-Electronics Review*, vol. 21, no. 1, p. 86 – 102, 2013, cited by: 49; All Open Access, Hybrid Gold Open Access.

[3] K. Kale, S. Pawar, and P. Dhulekar, "Moving object tracking using optical flow and motion vector estimation," in *2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, 2015, pp. 1–6.

[4] R. Gal, N. Kiryati, and N. Sochen, "Progress in the restoration of image sequences degraded by atmospheric turbulence," *Pattern Recognition Letters*, vol. 48, pp. 8–14, 2014, celebrating the life and work of Maria Petrou.

[5] H. Chao, Y. Gu, and M. Napolitano, "A survey of optical flow techniques for robotics navigation applications," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 73, no. 1-4, p. 361 – 372, 2014, cited by: 93.

[6] W. Enkelmann, "Obstacle detection by evaluation of optical flow fields from image sequences," *Image and Vision Computing*, vol. 9, no. 3, p. 160 – 168, 1991, cited by: 69.

[7] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, 2008.

[1] https://dsec.ifi.uzh.ch

[2] https://github.com/RD211/ev-optical-flow-comparative-study

[8] S. Shiba, Y. Aoki, and G. Gallego, *Secrets of Event-Based Optical Flow*. Springer Nature Switzerland, 2022, p. 628–645.

[9] A. Z. Zhu, D. Thakur, T. Ozaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The multivehicle stereo event camera dataset: An event camera dataset for 3d perception," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, p. 2032–2039, Jul. 2018.

[10] V. Brebion, J. Moreau, and F. Davoine, "Real-time optical flow for vehicular perception with low- and high-resolution event cameras," 2021.

[11] M. Gehrig, W. Aarents, D. Gehrig, and D. Scaramuzza, "Dsec: A stereo event camera dataset for driving scenarios," 2021.

[12] S. Shiba, Y. Klose, Y. Aoki, and G. Gallego, "Secrets of event-based optical flow, depth, and ego-motion by contrast maximization," *IEEE Trans. Pattern Anal. Mach. Intell. (T-PAMI)*, pp. 1–18, 2024.

[13] Y. Wu, F. Paredes-Vallés, and G. C. H. E. de Croon, "Lightweight event-based optical flow estimation via iterative deblurring," 2024.

[14] H. Liu, G. Chen, S. Qu, Y. Zhang, Z. Li, A. Knoll, and C. Jiang, "Tma: Temporal motion aggregation for event-based optical flow," 2023.

[15] Y. Li, Z. Huang, S. Chen, X. Shi, H. Li, H. Bao, Z. Cui, and G. Zhang, "Blinkflow: A dataset to push the limits of event-based optical flow estimation," 2023.

[16] R. Benosman, S.-H. Ieng, C. Clercq, C. Bartolozzi, and M. Srinivasan, "Asynchronous frameless event-based optical flow," *IEEE Transactions on Neural Natworks*, 01 2011.

[17] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision (ijcai)," vol. 81, 04 1981.

[18] M. Liu and T. Delbrück, "Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors," in *British Machine Vision Conference*, 2018.

[19] M. Liu and T. Delbruck, "Edflow: Event driven optical flow camera with keypoint detection and adaptive block matching," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 9, pp. 5776–5789, 2022.

[20] R. Benosman, C. Clercq, X. Lagorce, S.-H. Ieng, and C. Bartolozzi, "Event-based visual flow," *IEEE Transactions on Neural Networks*, vol. pp, p. 1, 11 2013.

[21] H. Akolkar, S. Ieng, and R. Benosman, "Real-time high speed motion prediction using fast aperture-robust event-driven visual flow," 2020.

[22] J. D. Adarve and R. Mahony, "A filter formulation for computing real time optical flow," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 1192–1199, 2016.

[23] C. Lee, A. K. Kosta, A. Z. Zhu, K. Chaney, K. Daniilidis, and K. Roy, "Spike-flownet: Event-based optical

flow estimation with energy-efficient hybrid neural networks," 2020.

[24] M. Gehrig, M. Millhäusler, D. Gehrig, and D. Scaramuzza, "E-raft: Dense optical flow from event cameras," 2021.

[25] A. Z. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Unsupervised event-based learning of optical flow, depth, and egomotion," 2018.

[26] F. Paredes-Vallés, K. Y. W. Scheper, C. D. Wagter, and G. C. H. E. de Croon, "Taming contrast maximization for learning sequential, low-latency, event-based optical flow," 2023.

[27] J. Hagenaars, F. Paredes-Valles, and G. de Croon, "Self-supervised learning of event-based optical flow with spiking neural networks," 2021.

[28] A. Zhu, L. Yuan, K. Chaney, and K. Daniilidis, "Evflownet: Self-supervised optical flow estimation for event-based cameras," in *Robotics: Science and Systems XIV*, ser. RSS2018. Robotics: Science and Systems Foundation, Jun. 2018.

[29] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[30] Z. Teed and J. Deng, "Raft: Recurrent all-pairs field transforms for optical flow," 2020.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[32] J. Nagata, Y. Sekikawa, and Y. Aoki, "Optical flow estimation by matching time surface with event-based cameras," *Sensors*, vol. 21, no. 4, 2021.

[33] S. Baker, S. Roth, D. Scharstein, M. J. Black, J. Lewis, and R. Szeliski, "A database and evaluation methodology for optical flow," in *2007 IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[34] S. Shiba, Y. Aoki, and G. Gallego, "Fast event-based optical flow estimation by triplet matching," *IEEE Signal Processing Letters*, vol. 29, pp. 2712–2716, 2022.

[35] Prophesee, *Metavision SDK*, n.d., version 4.6.1. [Online]. Available: https://docs.prophesee.ai/stable/algorithms/optical_flow.html

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[37] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019.

A Classical model-based approaches results

More traditional optical-flow estimation methods were also benchmarked (Table 5), but their accuracy was not competitive with the more recent model-based or learning-based approaches. However, the Time Gradient method [35], a modified version of plane fitting, achieved impressive runtime performance on DSEC which might make it useful in certain situations where performance is critical.

Table 5: Comparison of classical model-based approaches in terms of accuracy and inference time on the DSEC dataset. All implementations of these algorithms are from the Prophesee Metavision SDK [35] and all run exclusively on the CPU. The runtime is averaged over 800 optical flow maps creations.

	EPE ↓	$\%_{3PE}$ ↓	CPU Time
PlaneFitting [20]	22.6	86.5	320ms
TripletMatching [34]	12.6	84.0	107ms
TimeGradient [35]	15.8	86.0	40ms