Chair of Spacecraft Systems
Department of Aerospace and Geodesy
Technical University of Munich

Benchmark for Optical Flow algoritm using PyTorch in Python and C++ for Event Camera

Almudena Milans del Bosch – 03771709

Supervisor: Prof. Dr. phil. Alessandro Golkar – Advisor: Rafael Iliasov

Introduction

In recent years, the field of **computer vision** has witnessed remarkable advancements, particularly with the emergence of **Event Cameras**. These cameras, diverging from traditional image sensors, capture data asynchronously based on changes in pixel illumination rather than fixed intervals. This dynamic approach reduces energy consumption, enables real-time usage, and significantly enhances temporal resolution in information-rich areas of the image. In the context of the growing space sector, these cameras offer unique opportunities, facilitating efficient recognition, tracking, and monitoring tasks.

This project's focus is to optimize the implementation of several **optical flow algorithms** utilizing the **EVK3 Event Camera** with the **IMX636 HD Sensor** [1].

Optical Flow

Optical flow is defined as the **vector field** that describes the **perceived motion** of objects, surfaces and edges in a visual scene.

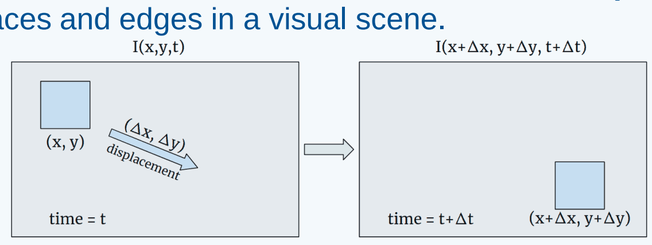


Fig 1. Optical Flow Definition [2]

Optical Flow Strategies

Sparse:
- **Packet-based:** Flow is generated on clusters of events.
- Complex but staged algorithm, leading to higher efficiency of high event-rate scenes.
- Approaches: Patented **Sparse Algorithm** of Prophesee [3].

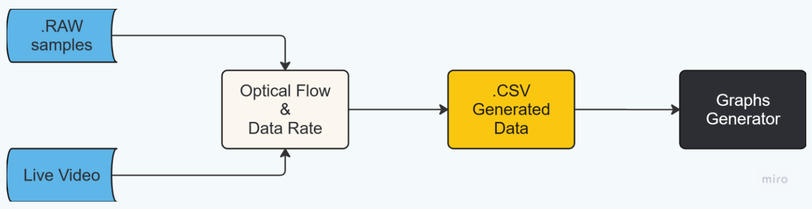
Dense:
- **Event-by-event-based:** Flow is computed for each event.
- Simple and efficient algorithm but run on all events hence is costly on high event-rate scenes.
- Approaches:
  - Triplet Matching Algorithm [4]
  - Plane Fitting Algorithm [5]

Software Set-Up

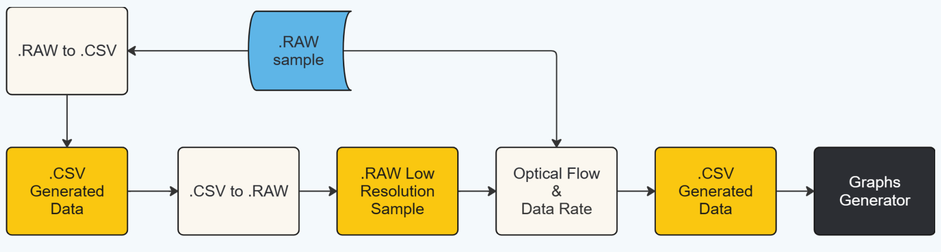
Computing Characteristics

OS	Windows 11
GPU	Intel® Core™ i7 – 10870H CPU @2.2GHz
RAM	16 GB

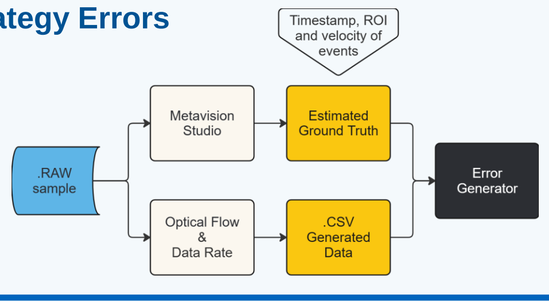
Recording and Real Time



Resolution



Optical Flow Strategy Errors



Results

The present study examines multiple sample recordings provided by Prophesee, as well as real-time recordings. The following results illustrate the distribution of time delay and RAM usage of the three benchmarked algorithm strategies.

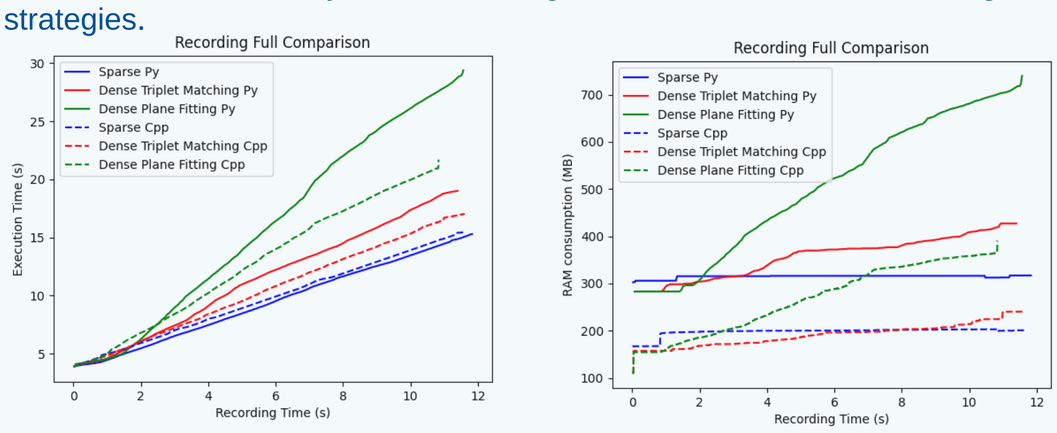


Fig 2. Real-time delay and RAM consumption

For the studied oscillating event-rate scene (fig. 2) ranging from 103,950 to 148,076 events per second, C++ consistently outperforms Python in terms of both lag and RAM consumption.

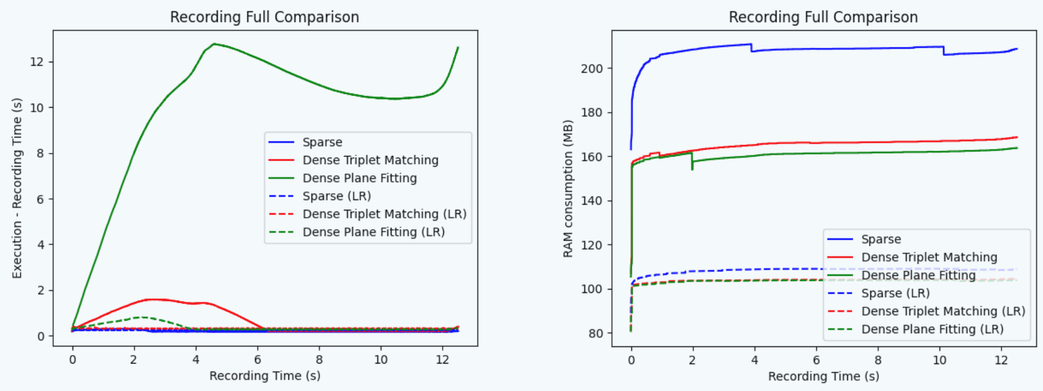


Fig 3. Resolution affection in C++ to delay and RAM consumption.

The high resolution of the captured scene results in lower performance as more events are being detected and processed. This tendency is followed both in Python and C++ approaches.

From the perspective of algorithm accuracy, several error metrics have been obtained:

		SP	TM	PF
Average Endpoint Error	$AEE = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(v_{x,i}-u_{x,i})^2+(v_{y,i}-u_{y,i})^2}$	483 px/s	1904 px/s	2425 px/s
Relative Endpoint Error	$REE = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\lvert u_i\rvert}\sqrt{(v_{x,i}-u_{x,i})^2+(v_{y,i}-u_{y,i})^2}$	36.5 %	143.9 %	183.3 %
Average Angular Error	$AAE = \frac{1}{N}\sum_{i=1}^{N} acos\frac{v_{x,i}u_{x,i}+v_{y,i}u_{y,i}}{\lvert v_i\rvert\lvert u_i\rvert}$	0.02 rad	2.20 rad	2.40 px/s

Ground truth has been estimated by filtering the corresponding events in the desired ROI with the correct timestamp.

Conclusions

The thesis achieved optimization of optical flow algorithms through comparative benchmarking between Python and C++. Sparse optical flow emerged as superior for accuracy and efficiency, particularly in real-time applications. Despite Python's flexibility, C++ excelled in efficiency and resource management, crucial for handling Event Camera data, as demonstrated in the comprehensive benchmarking.

Sources

[1] Metavision SDK Docs. (2024). EVK3 - VGA/320/HD. Metavision SDK Docs 4.5.2 documentation. EVK3 - VGA/320/HD — (prophesee.ai)

[2] Maxim Kuklin. Optical Flow in OpenCV (C++/Python). January 4, 2021 — (learnopencv.com)

[3] Method of processing a series of events received asynchronously from an array of pixels of an event-based light sensor. Patent Number: EP3694202A1

[4] Shintaro Shiba, Yoshimitsu Aoki, Guillermo Gallego. Fast Event-based Optical Flow Estimation by Triplet Matching. IEEE Signal Processing Letters, Vol. 29, pp. 2712-2716, 2022.

[5] Benosman, Ryad & Clercq, Charles & Lagorce, Xavier & Ieng, Sio-Hoi & Bartolozzi, Chiara. (2013). Event-Based Visual Flow. IEEE Transactions on Neural Networks. pp. 1. 10.1109/TNNLS.2013.2273537.