Laboratory of Computer Vision

Lab: Introduction to events cameras.

Author:

- **AMMAR KHODJA Rayane** 20225452@etud.univ-evry.fr
- **Daniel Shehroz khan SABIR JAMIL** 20235553@etud.univ-evry.fr

Supervised by:

- **Fabien Bonardi**

Master MMVAI November 2024.
Paris Saclay University.

Laboratoire de Recherche Paris Saclay University, IBISC Evry Val d'Essonne.

Abstract

This report is based on the research paper "Event-based Visual Flow" which introduces a new methodology for computing dense visual flow using the precise timings of spikes from an asynchronous event-based retina. The framework allows for the estimation of visual flow from the local properties of events' spatiotemporal space, with experimental results demonstrating its adequacy with high data sparseness and temporal resolution. The key advantages of using an event-based retina for computing visual flow are discussed, and the experimental results are presented to demonstrate the method's effectiveness. Overall, this research provides valuable insights for the development of event-based vision systems and their applications in various fields.

1 Introduction

Visual flow estimation is an active research area with applications in artificial vision and navigation. However, estimating dense flow fields on natural scenes in real-time remains challenging. Popular techniques can be categorized as energy-based, phase-based, correlation-based, or differential methods. While correlation-based methods perform well for stereo matching, differential methods are better suited for optical flow estimation. However, accuracy of differential techniques comes at the cost of computational complexity. Larger convolution kernels and temporal buffers improve accuracy but dramatically increase processing time, making real-time performance difficult. Thus, a trade-off exists between accuracy and efficiency. Recent work has focused on speeding up flow computation through parallelization and algorithmic improvements while retaining sufficient accuracy. However, additional progress is needed to achieve dense, precise flow estimation at high frame rates. This report surveys common visual flow techniques and recent advances in efficient and accurate flow estimation.

Important Note

- This whole work is based on the principles defined in the research paper of "Event-based Visual Flow" by Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng and Chiara Bartolozzi. However, the practical work is all 100% done by us, with the respected guidance thanks to Professor **Fabien Bonardi**.
- Attached with this repport you will find the "GoogleColab" notebook in python for the lab in visualizing as frames. Additionaly to that, you will find the "MATLAB" file for the execution as videos.

2 Theoretical part

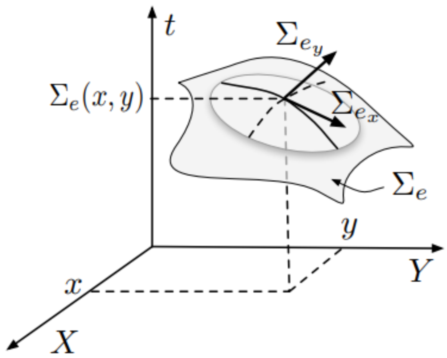


Figure 1: Illustration of the spatiotemporal surface defined by coactive events and the computation of the gradient for estimating visual flow.

The computation of visual flow is based on the spatiotemporal surface defined by coactive events. Each event is characterized by its position $p = (x, y)$ and its time t. The function Σ_e maps each position p to its corresponding time t, i.e., $\Sigma_e(p) = t$. Since time is a strictly increasing function, Σ_e is a non-zero derivatives surface at any point.

To estimate the visual flow, the gradient of the spatiotemporal surface is computed using a local differential approach. The gradient is given by:

$$\nabla\Sigma_e = \begin{pmatrix} \frac{\partial\Sigma_e}{\partial x} \\ \frac{\partial\Sigma_e}{\partial y} \end{pmatrix} \tag{1}$$

where $\frac{\partial\Sigma_e}{\partial x}$ and $\frac{\partial\Sigma_e}{\partial y}$ are the partial derivatives of Σ_e with respect to x and y, respectively. The gradient provides information about the orientation and amplitude of the visual flow at a given point.

To make the flow estimation robust against noise, a flow regularization process is added to the estimation. This process assumes local velocity constancy, which is equivalent to assuming that the spatiotemporal surface is locally planar. The slope of the fitted plane with respect to the time axis is directly proportional to the motion velocity. The regularization compensates for absent events in the neighborhood of active events where motion is being computed.

The theoretical part of the lab is crucial for understanding the methodology presented in the paper and for implementing it in practical applications. The computation of the gradient of the spatiotemporal surface is a key step in estimating the visual flow, and the flow regularization process helps to make the estimation robust against noise. Figure 3 illustrates the spatiotemporal surface defined by coactive events and the computation of the gradient for estimating visual flow.

What is the main focus of the methodology introduced in this paper?

The methodology introduced in this paper focuses on leveraging the unique characteristics of asynchronous event-based retinas to compute visual flow. Unlike traditional frame-based cameras, event-based retinas transmit less-redundant information about a visual scene in an asynchronous manner, and they rely on a paradigm of light acquisition that is radically different from most currently used frame-grabber technologies. The methodology aims to utilize the precise timing of spikes from these event-based retinas to compute visual flow with high accuracy and low computational cost, offering a new approach to processing visual information in demanding machine vision applications . This method does not rely on grey-levels or the integration of activity over long time intervals; instead, it uses each relative timing of changes of individual pixel's responses to visual stimuli as a computational input.

How do biological retinas and their artificial counterparts differ from traditional frame-grabber technologies?

Biological retinas and their artificial counterparts differ from traditional frame-grabber technologies in several key aspects:

1. **Asynchronous Operation**: Biological retinas and their artificial counterparts operate asynchronously, transmitting less-redundant information about a visual scene. In contrast, traditional frame-grabber technologies operate synchronously, capturing and processing frames at regular intervals .

2. **Data-Driven Paradigm**: Event-based retinas rely on a data-driven paradigm of light acquisition, where information is transmitted based on the occurrence of visual events, such as changes in log intensity, rather than capturing and processing entire frames at fixed time intervals .

3. **Redundancy Suppression**: Event-based retinas output compressed digital data in the form of events, removing redundancy and reducing latency compared to conventional imagers. This contrasts with traditional frame-grabber technologies, which capture and process entire frames, including redundant information .

4. **Temporal Resolution**: Event-based retinas offer high temporal resolution, with the precise timing of events conveyed with very low latency and accurate temporal resolution of 1s, enabling the computation of visual flow with micro-second accuracy .

Overall, the asynchronous and data-driven nature of biological retinas and their artificial counterparts, along with their high temporal resolution and redundancy suppression, distinguish them from traditional frame-grabber technologies and offer new opportunities for processing visual information in machine vision applications.

Local planes fitting algorithm on incoming events

The "Local planes fitting algorithm on incoming events" plays a crucial role in the flow regularization process for estimating visual flow based on incoming events. This algorithm aims to compensate for noise and absent events in the neighborhood of active events, thereby improving the robustness of the flow estimation.

The algorithm involves fitting local planes to the incoming events within a spatiotemporal window centered on each event. By assuming local velocity constancy, the algorithm exploits the planarity of the spatiotemporal surface defined by coactive events. This assumption allows for the estimation of motion velocity based on the slope of the fitted plane with respect to the time axis.

The use of local planes fitting helps to address the challenges posed by noise and data sparsity in event-based visual flow computation. By considering small spatiotemporal neighborhoods around individual events, the algorithm leverages the local properties of the event data to derive robust estimates of motion flow.

Furthermore, the algorithm provides an approximation of the timing of still non-active spatial locations, taking into account the asynchronous nature of the sensor and the non-idealities in event acquisition. This aspect is particularly important for maintaining accuracy in the presence of varying event densities and timing irregularities.

3 Practical work

3.1 Events cameras data VS usual camera frames

1. For this part you can check the google colab file attached, and MATLAB file attached.

2. Upon analyzing the provided visual data from the event camera, it is apparent that the overlapped events, depicted as yellow and green markers, correspond to the contours and edges where dynamic changes in the scene's luminosity occur. The yellow markers now represent positive polarity events, indicating an increase in light intensity, while the green markers signify negative polarity events, marking a decrease. These events are predominantly situated

Algorithm 1 Local planes fitting algorithm on incoming events

for all event $e(\mathbf{p}, t)$ **do** Define a spatio-temporal neighborhood Ω_e, centered on e of spatial dimensions $L \times L$ and duration $[t - \Delta t, t + \Delta t]$.
Initialization:

1. Apply least squares minimization to estimate the plane $\Pi = \begin{pmatrix} a & b & c & d \end{pmatrix}^T$ fitting all events $\tilde{e}_i\left(\mathbf{p}_i, t_i\right) \in \Omega_e$:

$$\tilde{\mathbf{\Pi}}_0 = \underset{\mathbf{\Pi} \in \mathbb{R}^4}{\operatorname{argmin}} \sum_i \left| \mathbf{\Pi}^T \begin{pmatrix} \mathbf{p}_i \\ t_i \\ 1 \end{pmatrix} \right|^2$$

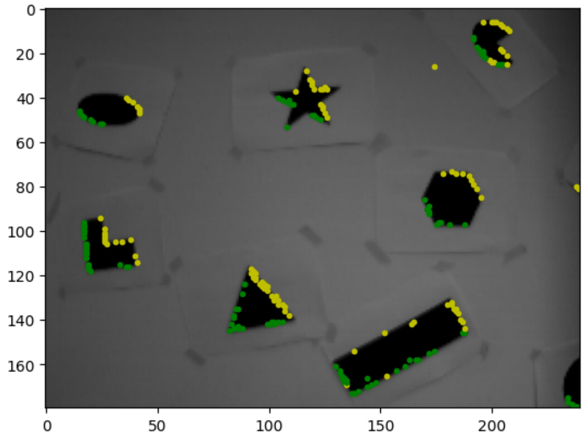
2. Set ϵ to some arbitrarily high value ($\sim 10^{-6}$).

while $\epsilon > th_1$ **do** Reject the $\tilde{e}_i \in \Omega_e$ if $\left| \tilde{\mathbf{\Pi}}_0^T \begin{pmatrix} \mathbf{p}_i \\ t_i \\ 1 \end{pmatrix} \right| > th_2$ (i.e., the event is too far from the plane) and apply Eq. 6 to estimate $\tilde{\Pi}$ with the non-rejected $\tilde{e}_i$ in Ω_e.
Set $\epsilon = \left\| \tilde{\mathbf{\Pi}} - \tilde{\mathbf{\Pi}}_0 \right\|$, then set $\tilde{\mathbf{\Pi}}_0 = \tilde{\mathbf{\Pi}}$.
Attribute to e the velocity defined by the fitted plane.
return $v_x(e), v_y(e)$.

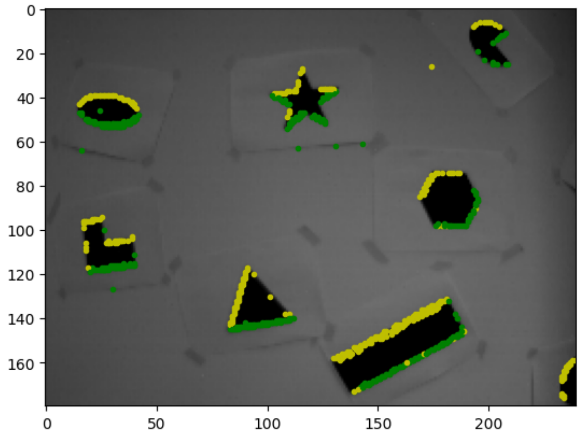
along the perimeters of distinct shapes within the frame, a characteristic trait of event cameras which excel in capturing high-contrast edges and motion. The static background, devoid of such markers, confirms the absence of significant intensity fluctuation, underscoring the camera's sensitivity to spatial and temporal variations. The distribution of these events offers insights into the temporal precision of the camera, capturing the essence of motion and light transitions within the observed scene.

3. Adjusting the temporal window for selecting events in the context of the event camera data profoundly influences the granularity and volume of captured events. A narrower temporal window yields a more refined selection of events, closely mirroring real-time changes in light intensity due to movement or illumination shifts, which is beneficial for capturing rapid dynamics. Conversely, expanding the temporal window encompasses a broader range of events, potentially introducing noise but also ensuring subtler variations in intensity are not overlooked. This trade-off can be observed in the frames where a denser clustering of events might indicate a longer window, capturing a prolonged activity sequence, while sparse distributions suggest a shorter window that isolates more transient changes. The choice of window thus directly affects the depiction of motion and contrast transitions within the visual data, offering either a more detailed or a more comprehensive view of the scene's temporal evolution.

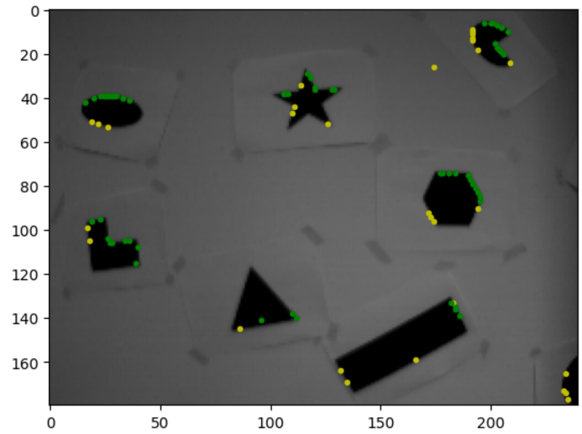
4. You can check the MATLAB files in order to see the whole videos for all the frames for each data set.



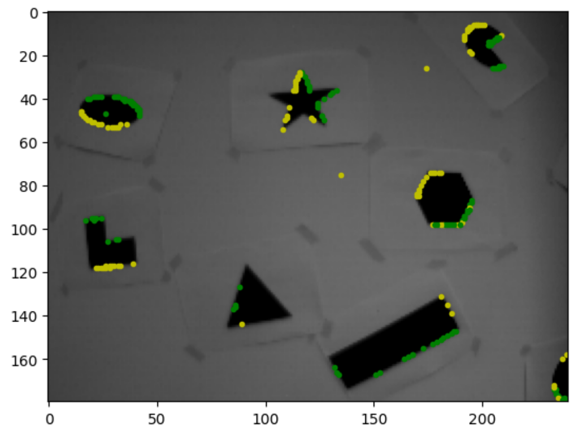
(a) The 1st frame



(b) The 2nd frame



(c) The 3rd frame



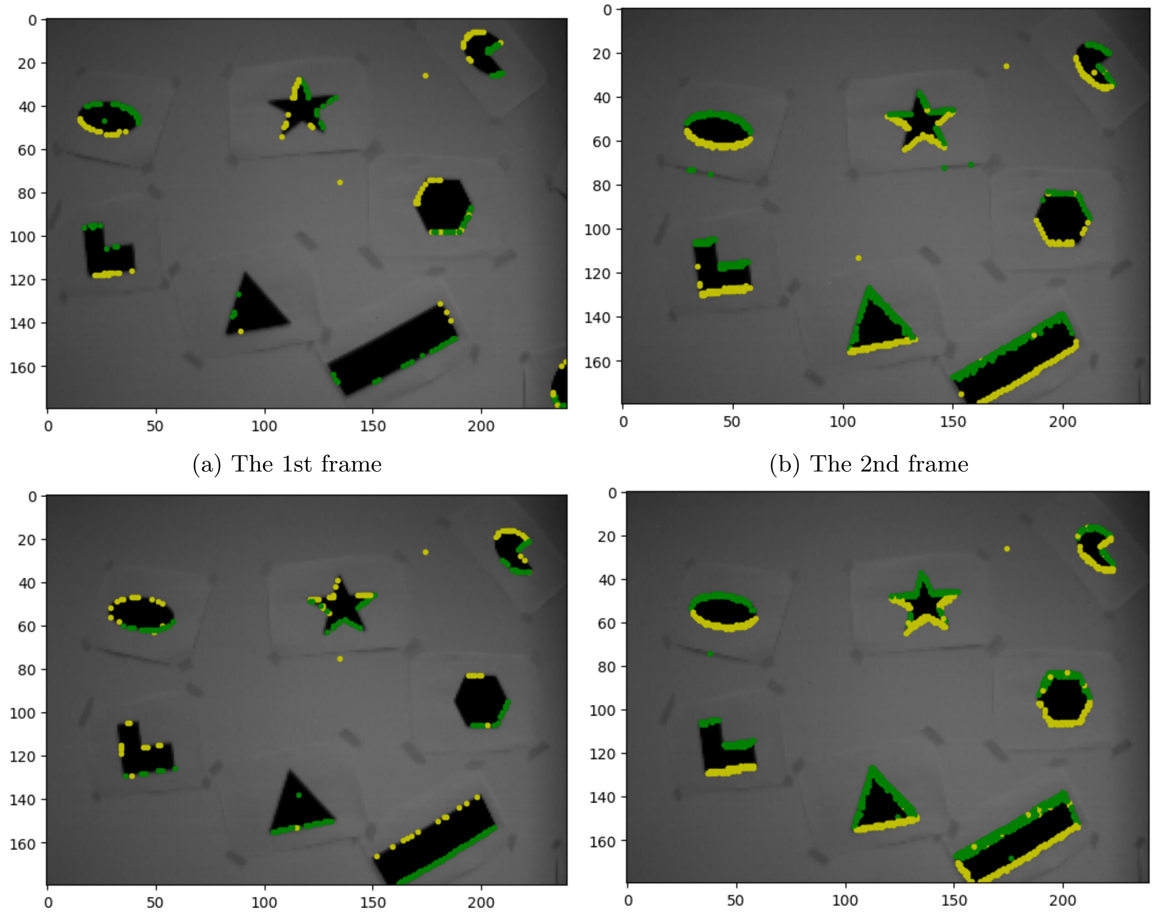
(d) The 4th frame

Figure 2: The results for the shapes_rotation dataset

3.2 Events data for objects tracking

1. With data from the first sequence, plot events in a 3D volume with x, y and t (timestamp) as the 3rd dimension:

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Extracting x, y coordinates and timestamps from the events for 3D plotting
x_coords = [event[1] for event in parsed_events]
y_coords = [event[2] for event in parsed_events]
timestamps = [event[0] for event in parsed_events]

# Setting up the 3D plot
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')

# Scatter plot for events in the 3D space
sc = ax.scatter(x_coords, y_coords, timestamps, c='g', marker='x')

# Labeling the axes
ax.set_xlabel('X␣coordinate')
ax.set_ylabel('Y␣coordinate')
ax.set_zlabel('Timestamp␣(s)')

# Title
plt.title('3D␣Plot␣of␣Events')
```



(a) The 1st frame

(b) The 2nd frame

(c) The 3rd frame

(d) The 4th frame

Figure 3: The results for the shapes_translation dataset

```
# Show the plot
plt.show()
```

2. **Note:**You can check the results in the attached code.

3. RANSAC Algorithm

Algorithm 2 RANSAC for Model Estimation

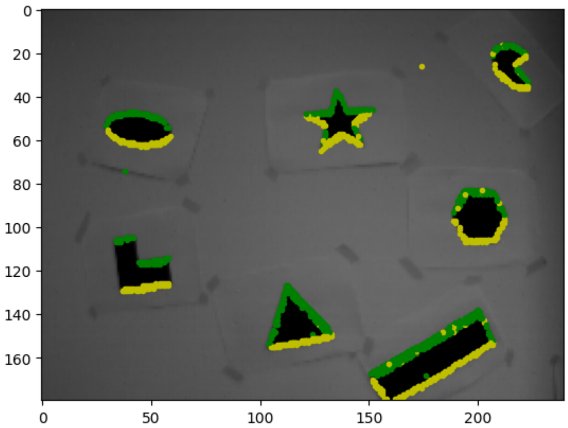
---

0 -1

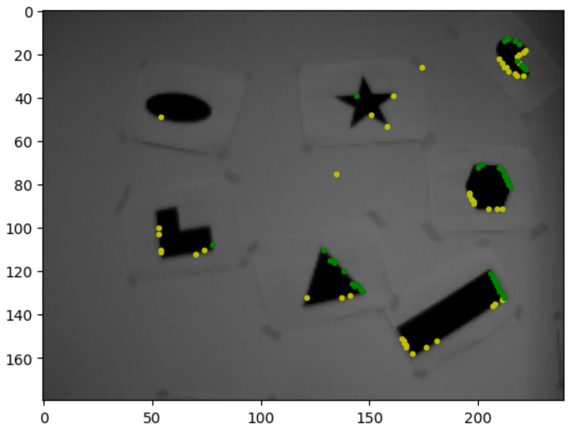
Observed data points D, Model parameters M, Threshold θ, Number of iterations N

4. **for** $i = 1$ N **do** Randomly sample a minimal subset Select a random subset $S \subset D$ of minimum size required for model fitting
Fit model to the subset Fit a model M to the data in subset S
Evaluate the model Compute the set of inliers I by applying the model to all data points and checking if they are within a threshold θ
Check model quality

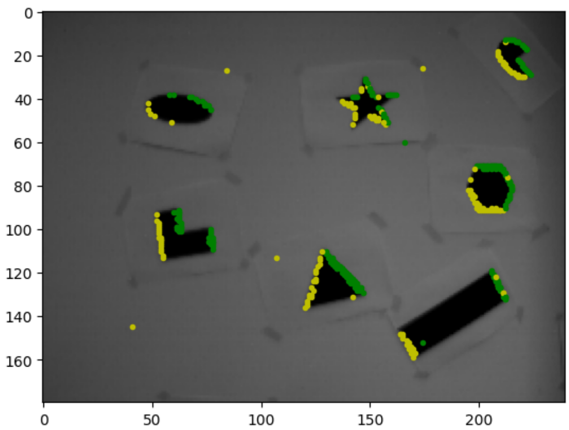
5. **if** Number of inliers in I is above a certain threshold **then** Re-estimate the model parameters using all inliers
return Best model parameters



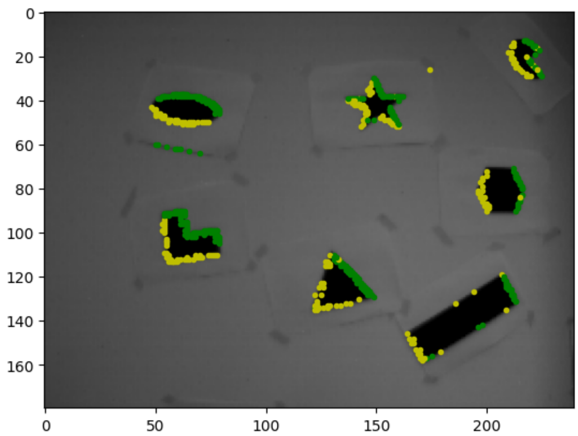
(a) The 1st frame



(b) The 2nd frame

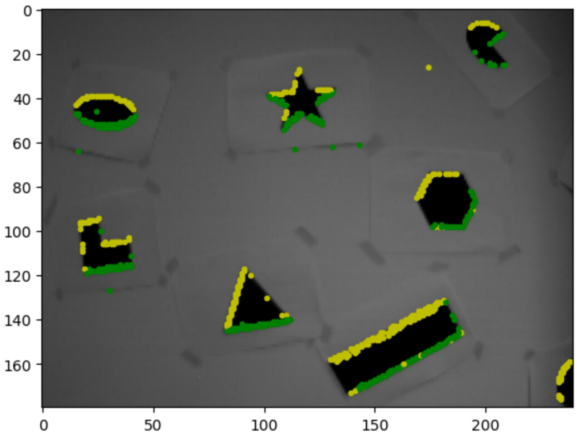


(c) The 3rd frame

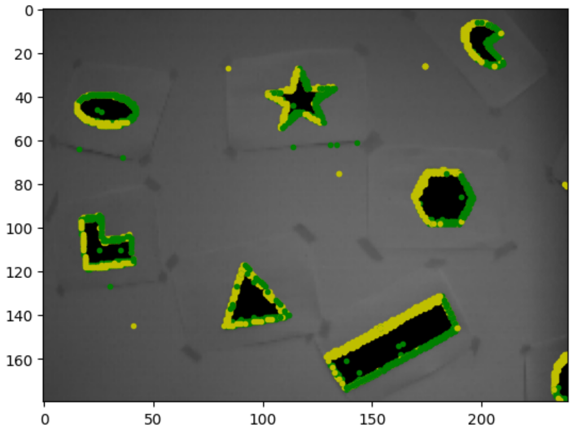


(d) The 4th frame

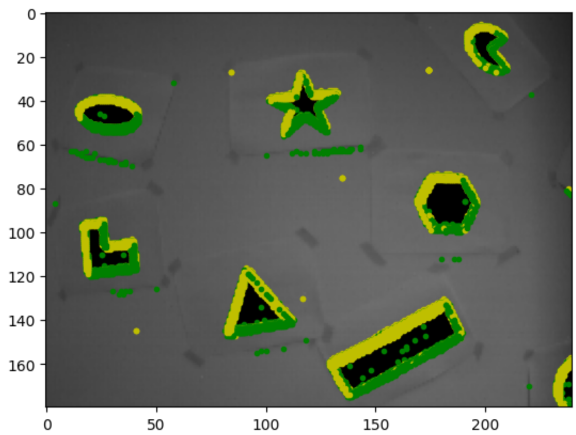
Figure 4: The results for the shapes_6dof dataset



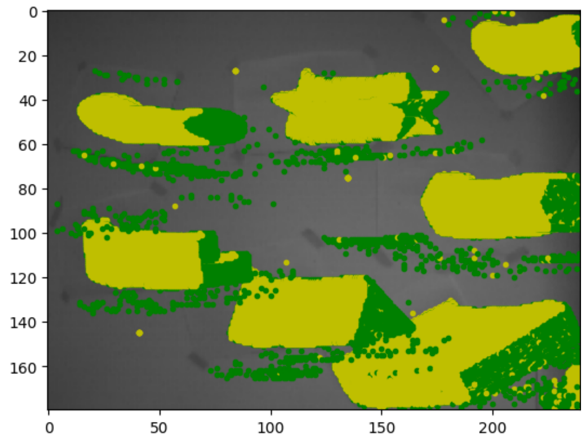
(a) Temporal window=0.01



(b) Temporal window=0.1



(c) Temporal window=0.5



(d) Temporal window=1.0

Figure 5: The results by comparing diffrent values of temporal window for 2nd frame of the shapes_rotaton dataset

3.3 Event-based visual flow

The stream of events from the silicon retina can be mathematically defined as follows: let $e(p,t) = (p,t)^T$ be a triplet giving the pixel location p and the time t of an event. The event stream can be represented as a set of such triplets, $E = \{e_1, e_2, ..., e_n\}$, where n is the number of events in the stream.

The motion flow at a given pixel p can be defined as the displacement of the pixel between two consecutive events, normalized by the time interval between the events. Let $v(p,t)$ be the motion flow at pixel p and time t. Then, we can define the motion flow as:

$v(p,t) = \frac{p_t - p_{t-1}}{t-(t-1)} = p_t - p_{t-1}$

where p_t and p_{t-1} are the pixel locations at times t and $t-1$, respectively.

3.4 Flow computation

The paper proposes a method for computing motion flow from the local properties of events' spatiotemporal space. The method involves computing the covariance matrix of the event locations in a local spatiotemporal neighborhood, and then using the eigenvectors of the covariance matrix to estimate the orientation and amplitude of the motion flow.

Let E_p be the set of events in a local spatiotemporal neighborhood of pixel p. The covariance matrix of E_p can be computed as:

$\Sigma_p = \frac{1}{|E_p|} \sum_{e_i \in E_p} (e_i - \bar{e}_p)(e_i - \bar{e}_p)^T$

where $\bar{e}_p$ is the mean of the event locations in E_p, and $|E_p|$ is the number of events in E_p.

The eigenvectors of Σ_p can be used to estimate the orientation and amplitude of the motion flow. Let λ_1 and λ_2 be the eigenvalues of Σ_p, and let u_1 and u_2 be the corresponding eigenvectors. Then, the orientation of the motion flow can be estimated as:

$\theta_p = \tan^{-1}\left(\frac{u_{1,y}}{u_{1,x}}\right)$

and the amplitude of the motion flow can be estimated as:

$a_p = \sqrt{\lambda_1^2 + \lambda_2^2}$

The paper also proposes a regularization method to improve the accuracy of the flow estimation. The regularization involves updating the covariance matrix Σ_p at each time step using a weighted average of the current covariance matrix and the covariance matrix of the previous time step:

$\Sigma_p \leftarrow \alpha \Sigma_p + (1-\alpha)\Sigma_{p,prev}$

where α is a regularization parameter, and $\Sigma_{p,prev}$ is the covariance matrix of E_p at the previous time step.

The paper presents experimental results demonstrating the effectiveness of the proposed method for computing motion flow from event-based retinas. The results show that the method is able to accurately estimate the orientation and amplitude of the motion flow in real-time, with micro-second accuracy and at very low computational cost.

Note: You can check the results in the attached code.

4 General Conclusion

In summary, the research paper proposes a method for computing motion flow from event-based retinas, utilizing the precise timing conveyed by the neuromorphic asynchronous event-based vision sensor. The method computes the covariance matrix of event locations in a local neighborhood to estimate the orientation and amplitude of the motion flow, and introduces a regularization process to improve the accuracy of flow estimation. Experimental results demonstrate the accuracy and real-time performance of the method, showing its capability to estimate motion flow with micro-second accuracy. Overall, the research paper provides a valuable contribution to the field of event-based vision and motion flow estimation, offering a method that leverages the unique properties of event-based retinas for real-time and low computational cost motion flow computation.