

SOFEA: A Non-iterative and Robust Optical Flow Estimation Algorithm for Dynamic Vision Sensors

Weng Fei Low, Zhi Gao, Cheng Xiang, and Bharath Ramesh*
The N.I Institute for Health, National University of Singapore

Email: bharath.ramesh03@u.nus.edu

Abstract

We introduce the single-shot optical flow estimation algorithm (SOFEA) to non-iteratively compute the continuous-time flow information of events produced from bio-inspired cameras such as the dynamic vision sensor (DVS). The output of a DVS is a stream of asynchronous spikes (“events”), transmitted at very minimal latency (1-10 μ s), caused by local brightness changes. Due to this unconventional output, a continuous representation of events over time is invaluable to most applications using the DVS. To this end, SOFEA consolidates the spatio-temporal information on the surface of active events for flow estimation in a single-shot manner, as opposed to iterative methods in the literature. In contrast to previous works, this is also the first principled method towards finding locally optimal set of neighboring events for plane fitting using an adaptation of Prim’s algorithm. Consequently, SOFEA produces flow estimates that are more accurate across a wide variety of scenes compared to state-of-the-art methods. A direct application of such flow estimation is rendering sharp event images using the set of active events at a given time, which is further demonstrated and compared to existing works (source code will be made available at our homepage after the review process).

1. Introduction

Neuromorphic sensors offer a complimentary approach to visual sensing and are increasingly explored for various applications, such as high-speed robotics [27, 28, 14, 9], robust feature detection and tracking [15, 19], object classification and tracking in low-power settings [25, 26, 30, 1, 16], conventional video synthesis [23, 7], space applications [11, 10], to name a burgeoning few.

Event cameras react to brightness changes sensed by an individual pixel, characterized by its location (x, y) , timestamp t and polarity p . The polarity specifies the brightness change direction as ON (increase denoted as +1) or OFF

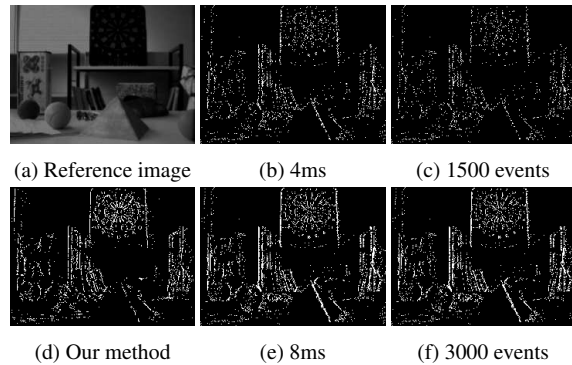


Figure 1: (a). The DVS is moved on a slider in front of objects at different depth [22]; (b) and (e). Time-slices of the event stream resulting in spatially irregular event accumulations; (c) and (f). Event-number slices which have the same drawback; (d). Optical flow estimated using SOFEA can be subsequently employed for lifetime estimation to produce crisp event stream images (best viewed on monitor).

(decrease denoted as -1). Additionally, these events are transmitted instantaneously with minimal latency in the order of 1-10 μ s. Thus, the output is a stream of asynchronous spikes (events) as opposed to a set of images captured at a constant rate by a standard CMOS camera. Due to this unconventional output, the design strategy of previous works has been to either implicitly or explicitly buffer a certain number of events for performing a given task.

A straightforward approach to processing DVS events is to accumulate them over a fixed time interval and then apply suitable frame-based methods. For instance, an events-to-frame approach was taken in [1, 30, 18] using contrived time periods (ranging between 5–66 ms) and were applied to specific scenarios of object tracking and stereo vision. During this events-to-frame conversion that introduces additional latency, the motion-dependent sensing aspect of the event camera is a major hindrance for choosing a fixed time-period for processing. Alternatively, a general way to

deal with the event camera output is to implicitly choose a time-period by buffering a certain number of past events [25, 14, 27]. However, choosing an ideal number of events to buffer while retaining critical image information is intricately dependent on the apparent object motion, scene complexity and spatial resolution of the sensor. Fig. 1 illustrates the drawback of these two ways of processing the event camera output.

In addition to the above ways of handling the event camera output, a third class of existing event-based works [23, 7] aim to render conventional video streams for traditional computer vision processing, albeit under challenging motion conditions where normal cameras fail to capture crisp images. These methods unsurprisingly revert to one of the main drawbacks of frame-based processing, *i.e.*, high power consumption [4].

In contrast to the above works, we explicitly model the set of active events in the spirit of [5] and compute continuous-time flow information. Previous works [5, 21, 2] employ iterative routines to estimate the visual flow, which are relatively slow and resource intensive in both software and hardware implementations for typically high event rates from the DVS. Thus, the motivation of our work is to devise a robust single-shot optical flow estimation algorithm (SOFEA) that allows for a broader range of downstream applications in a power-efficient manner. To this end, the two main theoretical contributions of this paper are:

- An efficient greedy algorithm to optimally select the neighboring events associated to an incoming event for accurate non-iterative local Surface of Active Events (SAE) plane fitting (Section 4.2).
- A mathematically simpler description of the local planar SAE, for efficient computation in hardware or software, is enabled by imposing an additional constraint that the incoming event lies on it (Section 4.3).

Our framework estimates the optical flow of each event, and thus, event stream images can be rendered at any given time instant using lifetime estimation, as shown and evaluated against [21] in the experiments section.

2. Related Work

The local plane fit approach is a widely adopted method [5] for event-based optical flow estimation. The single plane fit described in [29] is heavily prone to noise since it considers all neighbouring events with the same polarity. Therefore, to discard outlying data points, performing plane fitting *iteratively* is essential to improve accuracy [2, 21, 5]. The recent work [2] describes a hardware implementation by placing a 3×3 spatial constraint on the size of the neighborhood used for plane fitting, which limits the goodness-of-fit. Additionally, the incorporation of noise filtering and

old pixel thresholds [2] further limits the maximum speed of dynamics.

In contrast to the above plane fitting methods, [6] presented a variant of Lucas-Kanade flow estimation for event cameras. More recent approaches for optical flow estimation from event-based sensors include conventional block matching on adaptive time-sampled event stream images [20]. Reference [3] uses a variational approach that simultaneously estimates optical flow and image intensity as temporal contrast is generated due to a combination of both. Note that these methods are computationally intensive and have limited applications under low-power settings or real-time scenarios. In this regard, some works [17, 8] have advocated the use of spiking neural networks to process event-based data. Nonetheless, this area of research remains fairly unexplored and also requires dedicated hardware platforms.

Compared to the above works, our flow estimation approach neither depends on a temporal window (periodic/adaptive) nor is iterative [2, 21, 5]. Instead, we efficiently select the optimal neighbours of the incoming event and rely on a simple assumption that it lies on the local planar SAE for a single-shot flow estimation. Therefore, the proposed approach has potential for performing under low power settings in a small application specific IC, or to run real-time in software, or in embedded systems.

The remainder of the paper is organized as follows. The basic description of SAE and optical flow using plane fitting is given in Section 3. Next, Section 4 presents our proposed approach SOFEA for DVS. Section 5 presents the experimental results and comparison to existing works. Finally, the paper is concluded in Section 6.

3. Surface of Active Events and Optical Flow

An event e can be characterized by a 4-tuple (x, y, t, p) , where x, y represents the spatial location or position of the event in pixel coordinates, t represents the time of occurrence of the event and $p \in \{-1, +1\}$ represents the polarity of the event. On top of that, the event stream can be defined as a sequence of events $(e_n)_{n \in \mathbb{N}}$ such that $i < j \implies t_i \leq t_j$, where $e_i = (x_i, y_i, t_i, p_i)$, $e_j = (x_j, y_j, t_j, p_j)$.

Disregarding event polarity, each event carries information in the three-dimensional xyt spacetime domain. Consequently, the sequence of most recent events over the entire spatial domain, which is a subset of the event stream $(e_n)_{n \in \mathbb{N}}$, can be simply described by a spatio-temporal surface Σ_e . The spatio-temporal surface Σ_e is better known as the Surface of Active Events (SAE), as coined in [5]. The mathematical definition of the SAE is given by:

$$\begin{aligned} \Sigma_e : \mathbb{R}^2 &\rightarrow \mathbb{R} \\ \mathbf{p} &\mapsto \Sigma_e(\mathbf{p}) = t \end{aligned} \quad (1)$$

where $\mathbf{p} = [x \ y]^\top$.

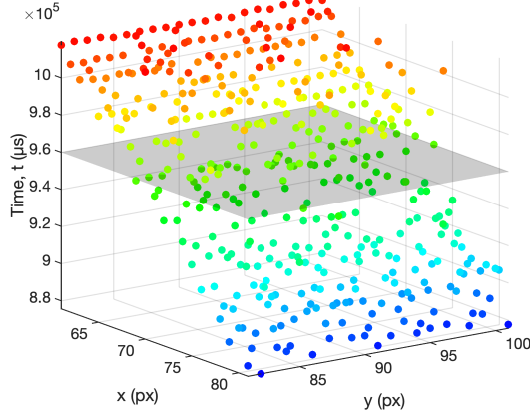


Figure 2: Spatio-temporal points on the Surface of Active Events (SAE) of a stripe translating in the negative x -axis direction. At a given time instant, the SAE can be used to generate the event stream image (grey plane), which is a binary representation of currently active pixels.

Note that Σ_e is a single-valued function of the event position \mathbf{p} . In other words, each point on the spatio-temporal surface indicates the occurrence or activation of an event at the particular position and time instant, given by its $[x \ y \ t]^\top$ coordinates. Thus, despite the asynchronous nature of the event stream, the exact position of all active or occurring events for any given time instant t can be easily determined from the corresponding time slice of the SAE, which we denote as an Event Stream Image. Figure 2 provides an illustration of a SAE and how an Event Stream Image can be acquired from the corresponding SAE.

A simple expression for Σ_e can be obtained by approximating the SAE as a plane in a local spatial neighborhood. In other words, the SAE is locally approximated by the first-order Taylor series expansion of Σ_e as follows:

$$\Sigma_e(\mathbf{p} + \Delta\mathbf{p}) \approx \Sigma_e(\mathbf{p}) + \nabla\Sigma_e(\mathbf{p})^\top \Delta\mathbf{p} \quad (2)$$

With the approximation above, the local planar SAE can be sufficiently described by a set of plane parameters $\mathbf{\Pi} = [a \ b \ c \ d]^\top$ as follows:

$$\begin{bmatrix} \mathbf{p} \\ t \\ 1 \end{bmatrix}^\top \mathbf{\Pi} = 0 \quad (3)$$

From the equation above, a closed form expression for Σ_e can also be derived as such:

$$\Sigma_e(\mathbf{p}) = \begin{bmatrix} -\frac{a}{c} & -\frac{b}{c} & -\frac{d}{c} \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \quad (4)$$

An interesting property of the SAE is that its spatial gradient $\nabla\Sigma_e(\mathbf{p}) = \begin{bmatrix} \frac{\partial\Sigma_e(\mathbf{p})}{\partial x} & \frac{\partial\Sigma_e(\mathbf{p})}{\partial y} \end{bmatrix}^\top$ encodes information of optical flow associated to the active events. For a local planar SAE, $\nabla\Sigma_e(\mathbf{p})$ can be further defined as $[-\frac{a}{c} \ -\frac{b}{c}]^\top$. For instance, suppose the stripe in Figure 2 was moving at half the speed, the magnitude of $\frac{\partial\Sigma_e(\mathbf{p})}{\partial x}$ would be twice as large, while $\frac{\partial\Sigma_e(\mathbf{p})}{\partial y}$ remains approximately zero. The mathematical relation of the normal flow with respect to $\nabla\Sigma_e(\mathbf{p})$ of a local planar SAE is given as such [29]:

$$\mathbf{v}_\perp = \frac{1}{\|\nabla\Sigma_e(\mathbf{p})\|} \nabla\hat{\Sigma}_e(\mathbf{p}) = -\frac{c}{a^2 + b^2} \begin{bmatrix} a \\ b \end{bmatrix} \quad (5)$$

where $\nabla\hat{\Sigma}_e(\mathbf{p}) = \frac{\nabla\Sigma_e(\mathbf{p})}{\|\nabla\Sigma_e(\mathbf{p})\|}$ is the unit vector of $\nabla\Sigma_e(\mathbf{p})$.

With the above analysis of $\nabla\Sigma_e(\mathbf{p})$, the local planar approximation of the SAE, as made in Equation 2, also implies an assumption of uniform normal flow across a local spatial neighborhood. This method of estimating the normal flow of events through the spatial gradient of its corresponding local planar SAE, as devised in [5], is widely known as the Local Plane Fit method [29].

4. Single-shot Optical Flow Estimation Algorithm (SOFEA) for DVS

In this section, we will introduce the Single-shot Optical Flow Estimation Algorithm (SOFEA) for DVS in detail. The motivation behind the algorithm is to devise a non-iterative and robust optical flow estimation method for event cameras that enables efficient hardware realization. This allows for a broader range of downstream applications in a power-efficient manner. Figure 3 shows the framework of SOFEA, which is based on the Local Plane Fit method described in Section 3.

The algorithm is executed for each incoming event from the event stream, with the current event being processed denoted as the Event-of-Interest (EOI) e_{EOI} . First, the EOI passes through a refractory filter, which contributes to a better conditioned SAE for subsequent processing. Then, we greedily select past spatially neighboring events that are (believed to be) associated to the EOI so that a local SAE plane fitting can be performed in a fast and non-iterative manner, while maintaining robustness of the plane fit. A final noise rejection filter, based on the goodness-of-fit, is employed before the normal flow estimate for the EOI is released to downstream applications. In the following subsections, we will address the details for each component of the framework given in Figure 3.

4.1. Refractory Filter

The refractory filter employed in SOFEA is closely related to the refractory period following a neuron action po-

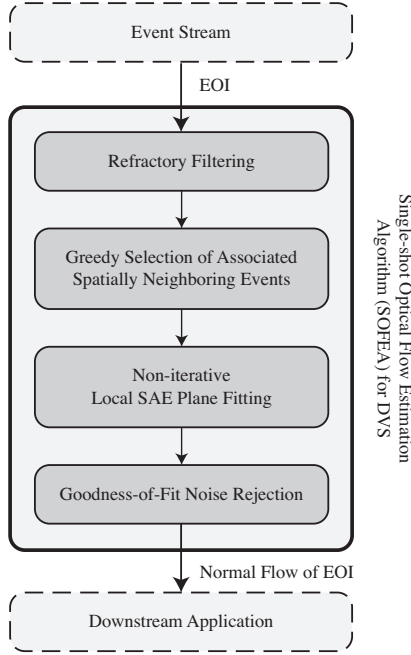


Figure 3: Framework of Single-shot Optical Flow Estimation Algorithm (SOFEA) for DVS.

tential [12]. As applied in [2, 25, 29], refractory filtering is performed by suppressing incoming events within a period of time after a past event has occurred at the same position. The period of time mentioned is known as the refractory period T_{rf} . Mathematically speaking, an event e_i will be filtered out if there exists an event e_j such that $x_i = x_j, y_i = y_j, t_i > t_j, t_i < t_j + T_{rf}$.

The incorporation of a refractory filter contributes to a better conditioned SAE for plane fitting [29]. Apart from achieving noise reduction, the refractory filter also induces invariance of the SAE to the scene dynamic range, to some extent, as it suppresses the burst of events following the first that occurs at the same position due to a change in contrast. However, this limits the maximum event occurrence rate at a particular position, and hence the maximum speed of dynamics [2]. With that said, the optimal value of T_{rf} depends on the scene dynamics and the DVS used, and may be appropriately set based on the intended application.

4.2. Greedy Selection of Associated Spatially Neighboring Events

To estimate the parameter Π of the local planar SAE associated to the EOI, events in the $L \times L$ (L is odd) spatial neighborhood are utilized to perform plane fitting. In contrast to [5], we only consider events occurred in the past,

including the EOI, for fitting the local planar SAE. This approach is also similarly adopted in [2, 21] so that no additional latency is introduced and no limits (both lower and upper bound) are implicitly imposed onto the speed of dynamics. With that, it is not a trivial task to distinguish spatially neighboring events of the EOI that truly belong to the local planar SAE, as some of them belong to that of the past event replaced by the EOI. Moreover, some events may be solely attributed to noise. In order to perform robust local SAE plane fitting in a single-shot manner, the set of spatially neighboring events used \mathcal{N} , where in practice is of a fixed size N_{nb} , have to be selected based on some optimality criteria.

Although there are no definitive criteria for the optimal set of neighboring events, which we will denote as \mathcal{N}^* , it can be observed that edges and natural structures that induce neighboring events eventually occur at time instants close to that of the EOI. Secondly, the optimal neighbours occur at positions that maintain spatial connectivity with the EOI, as well as other associated events. In addition, associated neighboring events usually have the same polarity as the EOI. Hence, we adopt the following as the criteria for \mathcal{N}^* of size N_{nb} :

1. Events in \mathcal{N}^* have a common polarity with the EOI.
2. Events in \mathcal{N}^* are located in the $L \times L$ spatial neighborhood about the EOI.
3. The EOI is not an event in \mathcal{N}^* .
4. Each event in \mathcal{N}^* is a 8-neighbor of the EOI or other events in \mathcal{N}^* .
5. The maximum timestamp difference between the EOI and events in \mathcal{N}^* is minimized. In other words, the minimum timestamp of events in \mathcal{N}^* is maximized, since the EOI has the largest timestamp.
6. Suppose Criterion 1 to 5 yield non-unique solutions, \mathcal{N}^* is then taken as one of them that gives the maximum sum of timestamps.

With the optimality criteria above, it can be proven by mathematical induction that the following greedy heuristic yields a locally optimal set of neighboring events \mathcal{N} (proof provided in the supplementary material):

At each stage, select the common polarity event with the next largest timestamp in the $L \times L$ spatial neighborhood about the EOI, which is also an 8-neighbor of the EOI or any previously selected event.

Figure 4 provides a graphical illustration of the greedy selection algorithm. An efficient implementation of the procedure is also given in Algorithm 1, which is an adaptation of Prim’s algorithm [24] for finding the minimum spanning

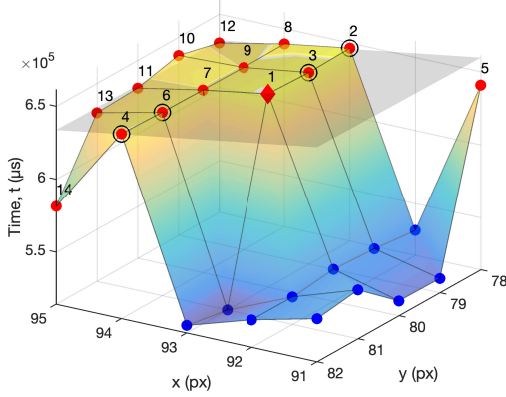


Figure 4: SAE in the 5×5 pixel spatial neighborhood about $e_{EOI} = (93 \text{ px}, 80 \text{ px}, 6.46697 \times 10^5 \mu s, -1)$, which is marked as a diamond. Events in \mathcal{L} , as defined in Algorithm 1, are marked as a sphere. Each event is colored in red or blue, indicating a negative or positive polarity respectively. Numbers placed above the events in red, which have the same polarity as e_{EOI} , indicate their rank when sorted based on their timestamp in descending order. The set of red events circled in black represents the locally optimal set of neighboring events $\bar{\mathcal{N}}$ of size $N_{nb} = 4$ associated to e_{EOI} . Events in $\bar{\mathcal{N}}$ are selected in the following order (each event is denoted by their rank): $3 \rightarrow 2 \rightarrow 6 \rightarrow 4$. The grey plane corresponds to the local planar SAE associated to e_{EOI} , as a result of non-iterative plane fitting with $\bar{\mathcal{N}}$ and e_{EOI} .

tree of a given connected graph. The heap data structure can be used to perform the operation on Line 5 and 6 of Algorithm 1 more effectively, as commonly done in implementations Prim's algorithm. Simplified sorting networks can also be used instead for efficient hardware realization.

The greedy selection algorithm functions on a best effort basis, because there is no guarantee that N_{nb} neighboring events are selected at the end of the process. We make use of this fact to perform additional noise rejection on the event stream. In other words, we reject the EOI and consider it as noise if less than N_{nb} neighboring events are selected for plane fitting.

4.3. Non-iterative Local SAE Plane Fitting

After the locally optimal set of neighboring events $\bar{\mathcal{N}}$ have been determined, the local planar SAE associated to the EOI can then be robustly fitted in a non-iterative manner. By examining Equation 3, it can be observed that there exists infinitely many Π that corresponds to the same particular plane. Hence, a unique set of parameters can be obtained by imposing an additional constraint on Π . Equa-

Algorithm 1 Efficient Greedy Selection of the Locally Optimal Set of Neighboring Events $\bar{\mathcal{N}}$ Associated to the EOI

```

1:  $\mathcal{L} \leftarrow \{ e \mid e \text{ is in the } L \times L \text{ spatial neighborhood} \\ \text{about } e_{EOI} \text{ and } e \neq e_{EOI} \}$ 
2:  $\mathcal{C} \leftarrow \{ e \mid e \text{ is an 8-neighbor of } e_{EOI} \text{ with the same} \\ \text{polarity} \}$ 
3:  $\bar{\mathcal{N}} \leftarrow \emptyset$ 
4: while  $|\bar{\mathcal{N}}| < N_{nb}$  and  $\mathcal{C} \neq \emptyset$  do
5:    $e_{new} \leftarrow e \in \mathcal{C}$  with the largest timestamp
6:    $\mathcal{C} \leftarrow \mathcal{C} \cup \{ e \mid e \text{ is a 8-neighbor of } e_{new} \text{ with the} \\ \text{same polarity, } e \notin \mathcal{C} \cup \bar{\mathcal{N}} \text{ and } \\ e \in \mathcal{L} \} \setminus e_{new}$ 
7:    $\bar{\mathcal{N}} \leftarrow \bar{\mathcal{N}} \cup \{ e_{new} \}$ 
8: end while
9: return  $\bar{\mathcal{N}}$ 

```

tion 3 can be rewritten as follows, given by the constraint of $c = -1$ [2]:

$$\begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix}^\top \Omega = t, \quad \Omega = \begin{bmatrix} \nabla \Sigma_e(\mathbf{p}) \\ d \end{bmatrix} \quad (6)$$

With $\bar{\mathcal{N}}$ and the EOI, the ordinary least squares estimate of Ω , denoted as Ω_{LS} , can be subsequently derived as follows:

$$\Omega_{LS} = \begin{bmatrix} \mathbf{p}_1^\top & 1 \\ \vdots & \vdots \\ \mathbf{p}_{N_{nb}+1}^\top & 1 \end{bmatrix}^\dagger \begin{bmatrix} t_1 \\ \vdots \\ t_{N_{nb}+1} \end{bmatrix} \quad (7)$$

where $e_i = (\mathbf{p}_i, t_i, p_i) \in \bar{\mathcal{N}} \cup \{ e_{EOI} \}$, $i \in [1, N_{nb} + 1]$ and \dagger represents the Moore-Penrose pseudo-inverse operation.

Note that solving the equation above involves a $(N_{nb} + 1) \times 3$ matrix pseudo-inverse as well as other relevant matrix operations. Along with the fact that millions of events are generated every second from the DVS, the operation becomes relatively slow and resource intensive in either software or hardware implementations.

On the other hand, a mathematically simpler description of the local planar SAE associated to the EOI can be employed for an efficient computation. It is obtained by imposing an additional constraint that the EOI must lie on the fitted local planar SAE, which is synonymous to the following constraint:

$$d = - \begin{bmatrix} \mathbf{p}_{EOI} \\ t_{EOI} \end{bmatrix}^\top \begin{bmatrix} a \\ b \\ c \end{bmatrix} = - \begin{bmatrix} \mathbf{p}_{EOI} \\ t_{EOI} \end{bmatrix}^\top \begin{bmatrix} \nabla \Sigma_e(\mathbf{p}) \\ -1 \end{bmatrix} \quad (8)$$

Substituting Equation 8 into Equation 6 gives the following simple description of the local planar SAE associated to the EOI, which only depends on the gradient $\nabla \Sigma_e(\mathbf{p})$ of dimension 2:

$$\Delta \mathbf{p}^\top \nabla \Sigma_e(\mathbf{p}) = \Delta t \quad (9)$$

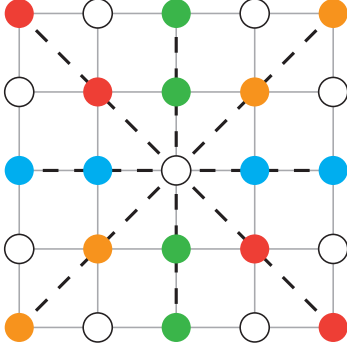


Figure 5: All 4 possible spatially collinear arrangements of events, visualized for a 5×5 pixel neighborhood. Each circle indicates a specific location in the local neighborhood, where the EOI is located on the central circle. The color of the circles (excluding white) reflect the particular collinear arrangement it is associated to. Thus, the set of events in $\bar{\mathcal{N}}$ are collinear if and only if all of them are located on circles of the same color, other than white.

where $\Delta \mathbf{p} = \mathbf{p}_{EOI} - \mathbf{p}$ and $\Delta t = t_{EOI} - t$.

The above simple description can also be similarly derived from Equation 2 with $\mathbf{p} = \mathbf{p}_{EOI}$. The ordinary least squares estimate of $\nabla \Sigma_e(\mathbf{p})$, denoted as $\nabla \Sigma_e(\mathbf{p})_{LS}$, can be obtained as follows:

$$\nabla \Sigma_e(\mathbf{p})_{LS} = \Delta P^\dagger \Delta \mathbf{t} \quad (10)$$

where:

$$\begin{aligned} \Delta P &= [\Delta \mathbf{p}_1 \quad \dots \quad \Delta \mathbf{p}_{N_{nb}}]^\top \\ \Delta \mathbf{t} &= [\Delta t_1 \quad \dots \quad \Delta t_{N_{nb}}]^\top \\ \mathbf{e}_i &= (\mathbf{p}_i, t_i, p_i) \in \bar{\mathcal{N}}, i \in [1, N_{nb}] \end{aligned}$$

For $\nabla \Sigma_e(\mathbf{p})_{LS}$ to be valid, ΔP^\dagger must exist and hence ΔP must be of full rank (rank 2). This requires the set of events in $\bar{\mathcal{N}}$ to be spatially non-collinear. Since the events in $\bar{\mathcal{N}}$ must remain spatially connected to the EOI (Criterion 4), collinearity can only take shape in 4 possible arrangements for any $L \geq 3$, as illustrated in Figure 5. With that, it can be easily shown that collinearity of $\bar{\mathcal{N}}$ can only happen if $N_{nb} \leq L - 1$.

To prevent the collinearity of $\bar{\mathcal{N}}$ when $N_{nb} \leq L - 1$, the following segment of pseudocode is inserted before line 7 of Algorithm 1 so that a collinearity check is incorporated:

```

1: if  $|\bar{\mathcal{N}}| = N_{nb} - 1$  and all events in  $\bar{\mathcal{N}} \cup \{e_{new}\}$ 
   are collinear then
2:   continue
3: end if

```

The collinearity check of $\bar{\mathcal{N}}$ can be performed efficiently by ensuring that not all events are situated on locations associated to the same collinear arrangement. This effectively causes the greedy selection algorithm to return the next best locally optimal set of events, in terms of Criterion 5 and 6, when all N_{nb} events selected are spatially collinear.

For an efficient hardware computation of $\nabla \Sigma_e(\mathbf{p})_{LS}$, the following can be done. From Equation 10, $\nabla \Sigma_e(\mathbf{p})_{LS}$ can be further expanded as follows:

$$\begin{aligned} \nabla \Sigma_e(\mathbf{p})_{LS} &= (\Delta P^\top \Delta P)^{-1} \Delta P^\top \Delta \mathbf{t} \\ &= \frac{1}{\det(\Delta P^\top \Delta P)} \text{adj}(\Delta P^\top \Delta P) \Delta P^\top \Delta \mathbf{t} \end{aligned} \quad (11)$$

where:

$$\begin{aligned} \Delta P^\top \Delta P &= \begin{bmatrix} \sum_{i=1}^{N_{nb}} \Delta x_i^2 & \sum_{i=1}^{N_{nb}} \Delta x_i \Delta y_i \\ \sum_{i=1}^{N_{nb}} \Delta y_i \Delta x_i & \sum_{i=1}^{N_{nb}} \Delta y_i^2 \end{bmatrix} \\ \Delta x &= x_{EOI} - x \\ \Delta y &= y_{EOI} - y \end{aligned}$$

Note that for a $L \times L$ (L is odd) spatial neighborhood about the EOI, Δx and Δy is strictly within the range $[-L_{\frac{1}{2}}, L_{\frac{1}{2}}]$, where $L_{\frac{1}{2}} = \frac{1}{2}(L - 1)$. With the above constraint, each of the product terms Δx^2 , $\Delta x \Delta y$ and Δy^2 in the entries of $\Delta P^\top \Delta P$ can be efficiently computed with the use of a Lookup Table (LUT) which stores all $\frac{1}{2}(L_{\frac{1}{2}} + 1)(L_{\frac{1}{2}} + 2)$ distinct pairwise products of integers in the range $[0, L_{\frac{1}{2}}]$. The sign of the product can be easily recovered from the signs of both operands. A total of $2L_{\frac{1}{2}} + 1$ entries can also be reduced from the LUT since pairwise products involving 0 and 1 can be trivially computed. With that, only $\frac{1}{2}L_{\frac{1}{2}}(L_{\frac{1}{2}} - 1)$ entries are required for the LUT. Therefore, each entry of $\Delta P^\top \Delta P$ can be computed with at most $N_{nb} - 1$ addition operations.

Apart from that, $\det(\Delta P^\top \Delta P)$ can also be calculated just using adders in a similar fashion with the aid of a LUT, suppose the range of values of the entries in $\Delta P^\top \Delta P$ is sufficiently small. With that, $(\Delta P^\top \Delta P)^{-1}$ can then be computed with a divider, where $\text{adj}(\Delta P^\top \Delta P)$ can be trivially deduced for a 2×2 matrix. Alternatively, $(\Delta P^\top \Delta P)^{-1}$ can be more directly obtained if it is practical to use a LUT to store all possible values of $(\Delta P^\top \Delta P)^{-1}$ corresponding to all possible combinations of $\bar{\mathcal{N}}$ with size N_{nb} , as similarly done in [2]. To minimize the number of multiplication operations, $\Delta P^\top \Delta \mathbf{t}$ is first computed, before it is multiplied by $(\Delta P^\top \Delta P)^{-1}$ to give the gradient of the local planar SAE $\nabla \Sigma_e(\mathbf{p})_{LS}$. Subsequently, the normal flow v_\perp can be derived from $\nabla \Sigma_e(\mathbf{p})_{LS}$ according to Equation 5.

4.4. Goodness-of-Fit Noise Rejection

In Section 4.2, we greedily select past spatially neighboring events that are believed to be associated to the EOI, according to an optimality criteria. Although a preliminary noise rejection is performed when less than N_{nb} events are selected at the end of the greedy selection process of $\bar{\mathcal{N}}$, it is still probable for the noisy EOI to evade such a measure by chance, especially when N_{nb} is small. Therefore, a final noise rejection is performed after local SAE plane fitting to ensure only robust normal flow estimates are released to downstream applications.

As we have constrained the EOI to be a point on the local planar SAE, we would expect quite a number of neighboring events with the same polarity to lie far away from the plane, given that the EOI is attributed to noise. Therefore, the final noise rejection measure incorporated is based on the assessment of the goodness-of-fit of the local planar SAE, which is similarly applied in [2, 5, 21].

We use the Δt residual of an event $e = (\mathbf{p}, t, p)$, which is given by the following, to indicate the distance of the event from the local planar SAE:

$$\hat{e} = \Delta t - \Delta \hat{t} = \hat{t} - t \quad (12)$$

where $\Delta \hat{t} = t_{EOI} - \hat{t} = \Delta \mathbf{p}^\top \nabla \Sigma_e(\mathbf{p})_{LS}$.

A neighboring event is said to support the fitted local planar SAE if the absolute of its Δt residual $|\hat{e}|$ is less than a constant threshold E (i.e. $|\hat{e}| < E$). With that, the local SAE plane fit, and hence the estimated $\nabla \Sigma_e(\mathbf{p})_{LS}$ and v_\perp , are considered to be valid given that there are at least N_{sp} neighboring support events with the same polarity as the EOI in the $L \times L$ spatial neighborhood about the EOI. More formally, the local SAE plane fit is valid if $|\mathcal{S}| \geq N_{sp}$, where the set of neighboring support events $\mathcal{S} = \{e \mid |\hat{e}| < E, p = p_{EOI} \text{ and } e \in \mathcal{L}\}$. As similarly shown in [21], N_{sp} is usually taken to be the rounded value of $\varepsilon(L^2 - 1)/2$, where ε denotes the nominal neighboring support ratio (inlier ratio) and $(L^2 - 1)/2$ represents the maximum neighboring support size.

5. Experiments

SOFEA is evaluated on various sequences used in previous works [21, 2, 20, 17] with ground truth flow estimates. Two metrics are used to evaluate different aspects of flow estimation, namely the Relative Endpoint Error EE_{rel} (magnitude of error in estimated velocities with respect to the ground truth in %) and Angular Error AE in space-time [29] (as the endpoint error does not differentiate between change in angle and speed). Sample mean and standard deviation values are also reported for EE_{rel} and AE.

A baseline SOFEA_{base} is used to provide a very close comparison to the benchmarks by matching parameters to the respective plane fitting methods (Table 1), and thereby

Table 1: Parameters of baseline SOFEA model for close evaluation to the respective plane fitting methods.

Sequence	T_{rf} (ms)	L (px)	N_{nb}	E (ms)	N_{sp}
stripes [21]	40	5	2	5	6
rotating_bar [2]	40	5	8	8	8

Table 2: Performance comparison of optical flow estimation methods on the stripes sequence.

Method	EE_{rel} (%)	AE (°)
[21]	81.03 ± 255.56	5.76 ± 13.59
SOFEA _{base}	24.53 ± 73.50	5.27 ± 11.64
SOFEA	14.46 ± 31.92	2.42 ± 10.63

examine the advantages of selecting optimal neighbours. The parameters used for the benchmark algorithms are themselves recommended values in their works. On the other hand, the full SOFEA model uses one set of parameters on all sequences, with $T_{rf} = 40$, $L = 7$, $N_{nb} = 16$, $E = 11$, and $N_{sp} = 15$. Note that SOFEA baseline model has not been used for the slider_hdr_far sequence as comparison is to a broader set of methods.

5.1. stripes sequence

This planar scene sequence from [21] generates DVS events at constant velocity, and thus ground truth can be obtained by inspection of constant time-interval accumulation of events for the stripes recording. Table 2 shows SOFEA significantly outperforming [21], in terms of EE_{rel} and AE. Similarly, the SOFEA_{base} has much better EE_{rel} compared to [21] but with par results using the AE metric. For this sequence, [21] uses $N = 5$ (matched to $L = 5$ in SOFEA_{base}), three events to fit (matched to $N_{nb} = 2 + \text{EOI}$), and $m = 7$ (matched to $N_{sp} = 6 + \text{EOI}$). Additionally, we statistically analyze the estimated lifetimes in Figure 6. Clearly, SOFEA has much closer correspondence to the ground truth lifetimes.

In contrast to [21], SOFEA is non-iterative as outlier rejection is inherent in the greedy selection of optimal neighbouring events to the EOI, and thus our method is less dependent on noise rejection. Moreover, the event stream image shows that critical visual information is retained by our method, as shown in Figure 7.

5.2. rotating_bar sequence

This sequence and the ground truth evaluations were reported in [2, 17, 5]. We compare SOFEA to the latest benchmark result [2] with recommended parameters for their iterative plane fitting method ($\Phi_{outlier} = 10\text{ms}$, $\Phi_{event} = 6$). For close comparison, SOFEA_{base} uses $L = 5$, as [2] uti-

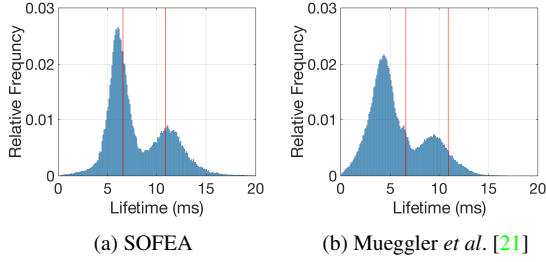


Figure 6: Empirical distribution of the estimated lifetime for the stripes sequence. The ground truth lifetimes corresponding to the close and far stripes are indicated by the left and right red vertical lines respectively.

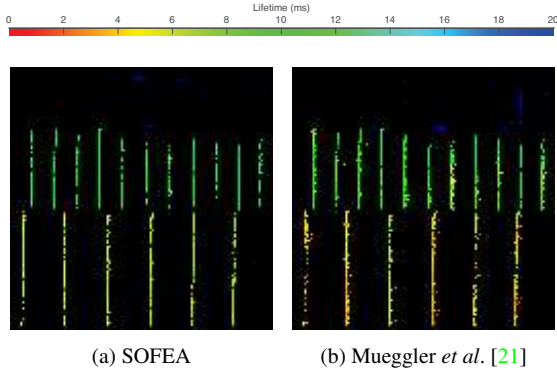


Figure 7: Event stream image obtained via lifetime estimation on the stripes sequence. Active events are highlighted with colors from red to blue indicating its estimated lifetime.

Table 3: Performance comparison of optical flow estimation methods on the `rotating_bar` sequence.

Method	EE _{rel} (%)	AE (°)
[2]	27.70 ± 33.23	14.92 ± 30.53
SOFEA _{base}	27.45 ± 55.35	7.02 ± 8.18
SOFEA	20.13 ± 26.62	6.21 ± 5.35

lizes a 5×5 pixel neighborhood by splitting it into multiple 3×3 regions for plane fitting.

Table 3 compares the performance of SOFEA against the iterative plane fitting method [2]. The results clearly demonstrate that the full SOFEA model produces much accurate flow estimates compared to the benchmark. On the other hand, SOFEA_{base} also has much better AE compared to [2] but with par results using the EE_{rel} metric. This indicates the strength of the greedy selection process while keeping all other parameters closely matched.

Table 4: Performance comparison of optical flow estimation methods on the `slider_hdr_far` sequence.

Method	EE _{rel} (%)	AE (°)
LP [5]	109.39 ± 83.82	16.99 ± 24.41
DS [13]	63.77 ± 58.91	21.46 ± 39.13
EBLK [6]	66.65 ± 17.59	13.52 ± 25.51
SOFEA	42.30 ± 87.57	16.73 ± 18.09

5.3. `slider_hdr_far` sequence

This sequence is a part of the Event Camera Dataset [22] consisting of a moving camera viewing a high-dynamic range scene. Note that since the DVS has a higher dynamic range of 120 dB (vs. 60 dB of standard camera), this sequence is of particular importance for evaluation. The focus is on comparison to the standard Local Plane Fit (LP) [5], and two other approaches, namely Direction Selective flow estimation (DS) [13] and Event-Based Lucas-Kanade (EBLK) [6].

Table 4 compares SOFEA against the above state-of-the-art works with standout results in terms of EE_{rel}. As for AE, EBLK performs the best while Local Plane Fit methods (SOFEA and LP) have a similar performance. Overall, SOFEA is still better than the EBLK method [6] due to the lower AE standard deviation. Note that SOFEA baseline model has not been used for this sequence as comparison is to a broader set of methods including LP works. In other words, SOFEA’s plane fitting is better due to the simpler local planar SAE model and the optimal selection of spatial neighbours using the greedy selection (Section 4.2).

6. Conclusion

We introduced SOFEA – a non-iterative and robust algorithm to estimate the optical flow of events. One of the main contributions of SOFEA include a simpler local planar SAE model by placing an additional constraint on the incoming event. Moreover, plane fitting was carried out in a single-shot manner by greedy selection of associated neighbouring events using an adaptation of Prim’s algorithm. In contrast to previous works, this is the first principled method for finding an optimal set of neighboring events for plane fitting. Our method consistently outperformed the state-of-the-art works across a wide variety of scenes using both the EE_{rel} and AE metrics. A direct application of such flow estimation is the rendering of crisp event stream images, which was demonstrated to retain critical information of the scene using SOFEA compared to a seminal work in the neuromorphic vision domain.

References

- [1] J. Acharya, A. U. Caycedo, V. R. Padala, R. R. Singh Sidhu, G. Orchard, B. Ramesh, and A. Basu. EBBIOT: A Low-complexity Tracking Algorithm for Surveillance in IoVT using Stationary Neuromorphic Vision Sensors. In *32nd IEEE International System-on-chip Conference*, 2019 (Accepted). 1
- [2] Myo Tun Aung, Rodney Teo, and Garrick Orchard. Event-based Plane-fitting Optical Flow for Dynamic Vision Sensors in FPGA. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, May 2018. 2, 4, 5, 6, 7, 8
- [3] P. Bardow, A. J. Davison, and S. Leutenegger. Simultaneous optical flow and intensity estimation from an event camera. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 884–892, 2016. 2
- [4] A. Basu, J. Acharya, T. Karnik, H. Liu, H. Li, J. Seo, and C. Song. Low-power, adaptive neuromorphic systems: Recent progress and future directions. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 8(1):6–27, 2018. 2
- [5] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi. Event-Based Visual Flow. *IEEE Transactions on Neural Networks and Learning Systems*, 25(2):407–417, Feb 2014. 2, 3, 4, 7, 8
- [6] Ryad Benosman, Sio-Hoi Ieng, Charles Clercq, Chiara Bartolozzi, and Mandyam Srinivasan. Asynchronous frameless event-based optical flow. *Neural Networks*, 27:32–37, 2012. 2, 8
- [7] C. Brandli, L. Muller, and T. Delbruck. Real-time, high-speed video decompression using a frame- and event-based davis sensor. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 686–689, June 2014. 1, 2
- [8] Tobias Brosch and Heiko Neumann. Event-based optical flow on neuromorphic hardware. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (Formerly BIONETICS)*, BICT’15, pages 551–558, 2016. 2
- [9] Andrea Censi and Davide Scaramuzza. Low-latency event-based visual odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 703–710. IEEE, 2014. 1
- [10] B. Cheung, M. Rutten, S. Davey, and G. Cohen. Probabilistic multi hypothesis tracker for an event based sensor. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1–8, July 2018. 1
- [11] Gregory Cohen, Saeed Afshar, Brittany Morreale, Travis Bessell, Andrew Wabnitz, Mark Rutten, and André van Schaik. Event-based sensing for space situational awareness. *The Journal of the Astronautical Sciences*, 66(2):125–141, Jun 2019. 1
- [12] P F Cranefield. Action potentials, afterpotentials, and arrhythmias. *Circulation Research*, 41(4):415–423, Oct 1977. 4
- [13] Tobi Delbruck. Frame-free dynamic digital vision. In *Proceedings of Intl. Symposium on Secure-Life Electronics, Advanced Electronics for Quality Life and Society*, pages 21–26, 2008. 8
- [14] Guillermo Gallego, Jon EA Lund, Elias Mueggler, Henri Rebecq, Tobi Delbruck, and Davide Scaramuzza. Event-based, 6-dof camera tracking for high-speed applications. *arXiv preprint arXiv:1607.03468*, 2016. 1, 2
- [15] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. Ekl: Asynchronous photometric feature tracking using events and frames. *International Journal of Computer Vision*, Aug 2019. 1
- [16] Arren Glover and Chiara Bartolozzi. Robust visual tracking with a freely-moving event camera. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3769–3776. IEEE, Sep 2017. 1
- [17] G. Haessig, A. Cassidy, R. Alvarez, R. Benosman, and G. Orchard. Spiking optical flow for event-based sensors using ibm’s trueneurosynaptic system. *IEEE Transactions on Biomedical Circuits and Systems*, 12(4):860–870, 2018. 2, 7
- [18] Jürgen Kogler, Christoph Sulzbachner, and Wilfried Kubinger. Bio-inspired stereo vision system with silicon retina imagers. In Mario Fritz, Bernt Schiele, and Justus H. Piater, editors, *Computer Vision Systems*, pages 174–183. Springer Berlin Heidelberg, 2009. 1
- [19] K. Li, D. Shi, Y. Zhang, R. Li, W. Qin, and R. Li. Feature tracking based on line segments with the dynamic and active-pixel vision sensor (davis). *IEEE Access*, 7:110874–110883, 2019. 1
- [20] Min Liu and Tobi Delbruck. Adaptive time-slice block-matching optical flow algorithm for dynamic vision sensors. In *BMVC*, 2018. 2, 7
- [21] E. Mueggler, C. Forster, N. Baumli, G. Gallego, and D. Scaramuzza. Lifetime estimation of events from dynamic vision sensors. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4874–4881, 2015. 2, 4, 7, 8
- [22] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and SLAM. *The International Journal of Robotics Research*, 36(2):142–149, 2017. 1, 8
- [23] Stefano Pini, Guido Borghi, Roberto Vezzani, and Rita Cucchiara. Video synthesis from intensity and event frames. In Elisa Ricci, Samuel Rota Bulò, Cees Snoek, Oswald Lanz, Stefano Messelodi, and Nicu Sebe, editors, *Image Analysis and Processing – ICIAP 2019*, pages 313–323, Cham, 2019. Springer International Publishing. 1, 2
- [24] R. C. Prim. Shortest Connection Networks And Some Generalizations. *Bell System Technical Journal*, 36(6):1389–1401, nov 1957. 4
- [25] B. Ramesh, H. Yang, G. M. Orchard, N. A. Le Thi, S. Zhang, and C. Xiang. DART: Distribution Aware Retinal Transform for Event-based Cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. 1, 2, 4
- [26] Bharath Ramesh, Shihao Zhang, Zhi Wei Lee, Zhi Gao, Garrick Orchard, and Cheng Xiang. Long-term object tracking with a moving event camera. In *BMVC*, 2018. 1

- [27] Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. Emvs : Event-based multi-view stereo. *Bmvc*, pages 1–11, 2016. [1](#), [2](#)
- [28] Henri Rebecq, Timo Horstschäfer, Guillermo Gallego, and Davide Scaramuzza. Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real-time. *IEEE Robotics and Automation Letters*, 2016. [1](#)
- [29] Bodo Rueckauer and Tobi Delbruck. Evaluation of event-based algorithms for optical flow with ground-truth from inertial measurement sensor. *Frontiers in Neuroscience*, 10:176, 2016. [2](#), [3](#), [4](#), [7](#)
- [30] A. Ussa, L. Della Vedova, V. Reddy Padala, D. Singla, J. Acharya, C. L. Zhang, G. Orchard, A. Basu, and B. Ramesh. A low-power end-to-end hybrid neuromorphic framework for surveillance applications. In *British Machine Vision Conference - Workshops*, 2019 (Accepted). [1](#)